

i-WAT: The Internet WAT System

– An Architecture for Maintaining Trust and Facilitating Peer-to-Peer
Barter Relationships –

Kenji Saito
ks91@sfc.wide.ad.jp

A Dissertation
Presented to the Faculty of the Graduate School
of
Keio University
in Candidacy for the Degree of
Doctor of Philosophy (Media and Governance)
January 18, 2006

Advisor:
Jun Murai

Committee Members:
Ikuyo Kaneko
Jiro Kokuryo
Masa Inakage

Abstract

i-WAT: The Internet WAT System
– An Architecture for Maintaining Trust and Facilitating Peer-to-Peer
Barter Relationships –
Kenji Saito
2006

Peer-to-peer barter currencies can be powerful tools for promoting exchanges and building sustainable relationships among selfish peers on the Internet.

i-WAT[74] is a proposed such currency based on the WAT System, a polycentric, real-life barter currency using *WAT tickets* as its media of exchange. Participants spontaneously issue and circulate the tickets as needed, whose values are backed up by chains of trust. *i*-WAT implements the tickets electronically by exchanging messages signed in OpenPGP[11].

Contributions of this research include the following:

1. Design of the total architecture of trust and fair exchange.
2. Construction, analysis and verification of the trust model of the system.
3. Construction, analysis and verification of the incentive models of the system, including those for *reduction/multiplicatoin-over-time* features.
4. Claim that the design of *i*-WAT is (group-)strategyproof, and is incentive-compatible as to counteraction against moral hazards. Threats caused by selfish peers taking advantage of the rules or their consequences are defused in *i*-WAT by the natural reactions of participants against those misbehaviors of others as they pursue their own benefits.
5. A reference implementation of *i*-WAT has been developed in the form of an XMPP[69, 70] (Extensible Messaging and Presence Protocol) instant messaging client.

The author has managed to put the currency system into practical use since June 2004.

要旨

iWAT: インターネット・ワットシステム-信用を維持し、ピア間のバーター取引を容易にするアーキテクチャ-

齊藤 賢爾

2006

ピア間で利用できるバーター通貨は、インターネット上の利己的なピア同士の交換を促進し、持続性のある関係を構築するための強力なツールとなり得る。

iWAT[74] は、そのような通貨となるべく、多中心的な補完通貨として実際に利用されているワットシステムに基づいて著者が提案している通貨システムである。ワットシステムでは、参加者は、その価値が信用の連鎖により担保されている、ワット券と呼ばれる交換媒体を用いる。参加者は、必要に応じて自発的に券を振り出し、それを流通させることができる。iWAT は、この券を、OpenPGP[11] により署名されたメッセージの交換により、電子的に実現する。

この研究は以下により、インターネット上での参加者間の良好な関係の構築と維持に貢献する:

1. 信用と公正な交換のためのトータル・アーキテクチャの設計。
2. システムの信用モデルの構築と分析、およびその検証。
3. システムのインセンティブ・モデルの構築と分析 (特に増減価型券の意味づけについて) およびその検証。
4. システムが、(グループ) 戦略耐性を持ち、かつ、モラルハザードに対して、インセンティブ整合的な耐性を持つという主張とその検証。利己的なピアが、ルールやその帰結を不当に利用して益を得ようとする脅威に対しては、iWAT では、参加者が自己の利益を追求すべく、そのような行為に対して自然に反応することにより対抗できる。
5. XMPP[69, 70] (Extensible Messaging and Presence Protocol) を利用するメッセージング・クライアントによるリファレンス実装。

著者は、2004年6月より iWAT を実用化し、実際に運用している。

Acknowledgments

This dissertation would not have been possible without the help of many people.

First and foremost, I would like to express my thankfulness to the scholars whose knowledges and guidances showed me the way. Above all, I would like to thank my primary advisor Jun Murai. I have learnt, and will continue to learn much from him, as he is a working example of pioneers, who has brought important changes to the societies. I would like to thank my advisors Ikuyo Kaneko, Jiro Kokuryo, Masa Inakage and Suguru Yamaguchi for their advices and enlightening conversations. I am especially grateful to Eiichi Morino, whose WAT System made all these possible.

Many people have been helping me through valuable advices and discussions on this research, as well as feedbacks on *wija* software. I would like to thank members of Gesell Research Society Japan and WIDE Project, who have tolerated yet underdeveloped human interface of my software, and willingly experimented on its new features. Many members of Inakage, Tokuda and Murai Laboratories at SFC (Shonan Fujisawa Campus), Keio University, have helped me improving *wija* software, and sometimes used it as the communication platform of their own researches, which worked as pacemakers of my development. Classes of my lecture “Introduction to Programming” in Spring and Fall 2005 at SFC have helped me improving *wija* and its LOGO programming environment.

Last, but not least, I would like to let my friends and family know that I am truly grateful to your supports, not just on this research, but on this wonderful life I have been spending with you in general.

Contents

Acknowledgments	i
Table of Contents	ii
List of Figures	ix
List of Tables	xiii
Nomenclature	xv
1 Introduction	1
1.1 Economics in the Presence of Replicators	1
1.1.1 Economics of the Star Trek Universe	1
1.1.2 Motivation for This Study	1
1.2 Dilemmas	2
1.2.1 Asymmetry in the Possible Outcomes	2
1.2.2 Tragedy of the Commons	3
1.2.3 Risks and Moral Hazards	3
1.3 Thesis Statement	4
1.4 Terminology	5
1.4.1 Exchange-Related Concepts	5
1.4.2 Safety-Related Concepts	7
2 Background	9
2.1 Historical Perspective	9
2.1.1 Complementary Currencies	9
2.1.2 Classification	9
2.1.3 Reduction Over Time	11
2.1.4 Multiplication Over Time	12
2.2 Peer-to-Peer Network	13
2.2.1 Concept	13
2.2.2 Peer-to-Peer Barter Currencies	14
2.3 Relevant Theories	14
2.3.1 Distributed Algorithmic Mechanism Design	14

2.3.2	Digital Signature	16
2.3.3	Web of Trust	17
2.3.4	PGP Trust Model	18
3	Problem Statements	19
3.1	Requirements	19
3.1.1	Autonomy	19
3.1.2	Safety	20
3.1.3	Integration	21
3.2	The WAT System	21
3.2.1	Overview	21
3.2.2	Distinctive Features of the WAT System	23
3.3	Problems in Electronizing the WAT System	23
3.4	<i>i</i> -WAT: the Internet WAT System	24
3.4.1	Overview	24
3.4.2	Conditions	25
3.4.3	Protocol	27
4	Theory	31
4.1	Overview	31
4.1.1	Approaches to Autonomy	31
4.1.2	Approaches to Safety	32
4.1.3	Approaches to Integration	34
4.2	Total Architecture	35
4.3	<i>i</i> -WAT and the PGP Trust Model	36
4.3.1	<i>i</i> -WAT Trust Model	36
4.3.2	Spinning the Web of Trust – Preconditions	36
4.3.3	Spinning the Web of Trust – Case Studies	37
4.3.4	Justification of the Preconditions	41
4.4	General Incentive Model	41
4.4.1	Generalized Ticket Value	41
4.4.2	Notations and Preconditions	42
4.4.3	Game Trees	44
4.5	ROT: Reduction Over Time	46
4.5.1	Concept	46
4.5.2	Incentive-Compatibility of the Design	47
4.6	MOT: Multiplication Over Time	51
4.6.1	Concept	51
4.6.2	Incentive-Compatibility of the Design	51
4.7	Strategies and Moral Hazards	53
4.7.1	Overview	53
4.7.2	Safety and Risks	55
4.7.3	Sloppy Key Management	56
4.7.4	Excessive Issuing	57

4.7.5	Lazy Approval	57
4.7.6	Defaults	57
4.7.7	Empty Promise	57
4.7.8	Collusions	58
4.8	Distributed Auditing	58
4.9	Public Key Exchange	60
4.9.1	Overview	60
4.9.2	Propagation of Signatures	60
4.9.3	Support for the preconditional properties	61
4.10	Extension Mechanism	61
4.10.1	Currency Name Space	61
4.10.2	Currency Semantics	61
4.11	Coexistence with Existing Currencies	63
4.11.1	Exchange Points and Translation Mechanism	63
4.11.2	Internetworking with an MCS	64
4.12	New Economic Order (NEO)	65
5	Practice	67
5.1	Reference Implementation	67
5.1.1	Overview	67
5.1.2	Outline of Jabber/XMPP	67
5.1.3	<i>wija</i>	69
5.1.4	<i>wijapo</i>	71
5.1.5	OMELETS	72
5.1.6	Public Relations	72
5.2	Implementation Issues	73
5.2.1	Implementation of <i>wija</i>	73
5.2.2	Implementation of <i>wijapo</i>	74
5.2.3	Implementation of <i>i-WAT</i>	75
5.3	Experiments	76
5.3.1	Overview	76
5.3.2	Preliminary Deployment of <i>i-WAT</i>	77
5.3.3	WIDE Hours	77
5.3.4	MANA	78
5.3.5	Public Releases of <i>wija</i>	79
5.3.6	Vegetable Trading	79
5.4	Simulation	83
5.4.1	Principles	83
5.4.2	The World	83
5.4.3	Repositories, Production and Consumption	84
5.4.4	Currencies	85
5.4.5	Welfare	86
5.4.6	Balance	86
5.4.7	Trades	86

5.4.8	Bankruptcy	86
6	Results	89
6.1	Simulated Results	89
6.1.1	Mass-Market MCS	89
6.1.2	Small-World MCS	90
6.1.3	The WAT System	92
6.1.4	Comparative Study on <i>i</i> -WAT (regular tickets)	93
6.1.5	Comparative Study on <i>i</i> -WAT (variance over time)	102
6.1.6	Economics in the Presence of Replicators	111
6.2	Results from Experiments	120
6.2.1	WIDE Hours	120
6.2.2	MANA	121
6.2.3	Vegetable Trading	121
6.3	Deployment of the Reference Implementation	139
6.3.1	Participants and Usage	139
6.3.2	Statistics	139
6.3.3	Approval Behaviors	142
6.4	Cases	144
6.4.1	General Purchasing	144
6.4.2	Supporting by ROT	144
6.4.3	Barter YEN and Dollar	145
6.5	Incidents	145
6.5.1	Security Incidents	145
6.5.2	Availability Incidents	147
7	Discussion	149
7.1	Meaning of the Results	149
7.1.1	Non-Intuitive Consequences	149
7.1.2	Moral Hazards and Counteraction	150
7.1.3	NEO and Motivations for Creations	150
7.1.4	Implied Institutional Changes	150
7.1.5	Differences between Models and Practices	151
7.1.6	Gender and Age-Distributions of Users	151
7.1.7	Implications of the Experimental Outcomes	153
7.2	Related Work	153
7.2.1	Overview	153
7.2.2	Centralized Debt-oriented Currencies	153
7.2.3	Decentralized Debt-oriented Currencies	154
7.2.4	Decentralized Labor-oriented Currencies	155
7.2.5	Incentive Techniques	155
7.3	Comparison of Trust Models	155
7.3.1	Comparison with Geek Credit/Ripple Trust Model	155
7.3.2	Comparison with Karma Trust Model	156

8	Conclusions	159
9	Recommendations	163
9.1	Research Plans	163
9.2	Unimplemented Features	163
9.2.1	Distributed Auditing	163
9.2.2	Privacy Support	164
9.2.3	More Public Key Management Support	164
9.2.4	Enforcement of the Security Rule	164
9.2.5	Automatic Backup and Distributed Restore	164
9.3	Application: Distributed Computing	165
9.4	Application: Sharing Creative Works	165
9.5	Application: Post-Catastrophic Recovery	165
9.5.1	Principles and Requirements	165
9.5.2	WAT/ <i>i</i> -WAT for Post-Catastrophic Recovery	166
9.5.3	Implementation of the Proposed Model	168
	Afterword	169
A	Protocols	171
A.1	Public Key Exchange Protocol	171
A.1.1	Request	171
A.1.2	Public Key Transfer	171
A.2	<i>i</i> -WAT Protocol	172
A.2.1	Issuing	172
A.2.2	Circulation	174
A.2.3	Redemption	178
A.3	Hypertext Sharing Protocol	180
A.3.1	Bookmark Transfer	180
A.3.2	Hypertext Transfer	180
B	Simulator	183
B.1	Acquisition	183
B.2	Data Description (XML)	183
B.3	Archetypes	184
B.3.1	DefaultParticipant (<i>i</i> -WAT user)	184
B.3.2	RedeemingParticipant	186
B.3.3	StretchingParticipant	186
B.3.4	SelectiveParticipant	186
B.3.5	OriginalWATParticipant	187
B.3.6	BankingParticipant	187
B.3.7	GlobalMarketBankingParticipant	187
B.3.8	BankingWATParticipant	187
B.3.9	PlaceboParticipant	188

B.3.10	PlaceboRedeemingParticipant	188
B.3.11	PlaceboStretchingParticipant	188
B.3.12	PlaceboSelectiveParticipant	188
B.3.13	SemiOptimizedParticipant	188
B.3.14	OptimizedParticipant	189
B.4	Command	189
B.5	Output Files and Formatting	189
C	Descriptive Replies to Questionnaires	193
C.1	WIDE Hours	193
C.1.1	Replies to Question 14 (Impression)	193
C.1.2	Replies to Question 17 (Interfaces)	194
C.1.3	Replies to Question 20 (Usage)	195
C.1.4	Replies to Question 21 (Improvements)	196
C.2	Vegetable Trading	197
C.2.1	Replies to Question 4 (Usefulness)	197
C.2.2	Replies to Question 5 (Suggestions or Other Thoughts)	198
	Bibliography	202

List of Figures

1.1	A roundabout and its incentive mechanism	4
1.2	A barter with or without currency	6
2.1	A sample of stamp scrip used in Mason City, Iowa, in 1930's .	11
2.2	A sample of calendar money in Salt Lake City, Utah, in 1930's	12
2.3	The model of mechanism design	15
3.1	Three stages of trading with a WAT ticket	22
3.2	Signature chain in an <i>i</i> -WAT ticket	25
3.3	Visualized signature chain in the reference implementation . .	25
3.4	State machine of a WAT/ <i>i</i> -WAT ticket	26
3.5	<i>i</i> -WAT transaction 1: issuing	28
3.6	<i>i</i> -WAT transaction 2: circulation	29
3.7	<i>i</i> -WAT transaction 3: redemption	29
4.1	Five-layer model	35
4.2	<i>i</i> -WAT trust model	36
4.3	Game tree for issuing a ticket	44
4.4	Game tree for circulating a ticket	45
4.5	Meaning of a <i>reduction</i> ticket	46
4.6	Game tree for issuing a <i>reduction</i> ticket	48
4.7	Game tree for circulating a <i>reduction</i> ticket	49
4.8	Meaning of a <i>multiplication</i> ticket	52
4.9	Game tree for issuing a <i>multiplication</i> ticket	53
4.10	Game tree for circulating a <i>multiplication</i> ticket	54
4.11	An example of an <i>i</i> -WAT ticket data	62
4.12	An example of an extension to <i>i</i> -WAT currency	62
4.13	Exchanging <i>i</i> -WAT tickets among different currencies	63
4.14	Exchanging MCS-based WIDE Hours outside its members . .	64
5.1	Overview of communication in Jabber/XMPP	69
5.2	Screenshots of <i>wija</i> and its plug-ins	70
5.3	<i>i</i> -WAT book	71
5.4	<i>wija</i> (left) and <i>i</i> -WAT (right) top pages	72

5.5	WIDE Hours (left) and MANA (right) ranking pages	78
5.6	Example: <i>i</i> -WAT version of <i>WIDE Hours</i> in circulation	79
5.7	Issuing in Vegetable Trading	81
5.8	Circulation in Vegetable Trading	81
5.9	Main (left) and signing (right) screens of Vegetable Trading	82
5.10	Link distributions in the small worlds	84
5.11	Sample initial world of population = 100	84
6.1	Welfare distributions in mass-market MCS	90
6.2	Welfare distributions in mass-market MCS with whitewashers	91
6.3	Population, welfare and rate of failures	91
6.4	Welfare distributions in small-world MCS	92
6.5	Welfare distributions in small-world MCS with whitewashers	93
6.6	Welfare distributions in the WAT System	94
6.7	Welfare distributions in the WAT System with whitewashers	94
6.8	Welfare distribution in <i>i</i> -WAT (bankruptcy rate: 0.002)	95
6.9	Welfare distribution in <i>i</i> -WAT without evasive actions (1)	96
6.10	Welfare distribution in <i>i</i> -WAT without evasive actions (2)	97
6.11	Welfare distribution in <i>i</i> -WAT with EV1	97
6.12	Welfare distribution in <i>i</i> -WAT with EV1 and EV2	98
6.13	Competing welfare distributions with or without EV2	99
6.14	Distributions of chain lengths with evasive actions	99
6.15	Welfare distribution in <i>i</i> -WAT with EV1, EV2 and EV3	100
6.16	Frequencies of trade types w/ or w/o evasive actions	101
6.17	Welfare distributions in <i>i</i> -WAT	101
6.18	Welfare distributions in <i>i</i> -WAT with whitewashers (1)	102
6.19	Welfare distributions in <i>i</i> -WAT with whitewashers (2)	103
6.20	Total debt with or without evasive actions	103
6.21	Welfare distribution in <i>i</i> -WAT (population: 500)	104
6.22	Welfare distributions with whitewashers (population: 500)	104
6.23	Over-time rates and mean welfare	105
6.24	Total debt with different over-time rates	106
6.25	Reduction ratios and mean welfare	106
6.26	Multiplication ratios and mean welfare	107
6.27	Distributions of chain lengths for <i>reduction</i> tickets	108
6.28	Competing welfare distributions with <i>reduction</i> tickets	109
6.29	Distributions of chain lengths for <i>multiplication</i> tickets	109
6.30	Competing welfare distributions with <i>multiplication</i> tickets	110
6.31	Welfare distributions with 10% ROT whitewashers	110
6.32	Welfare distributions with 20% ROT whitewashers	111
6.33	Welfare distributions with 10% MOT whitewashers	112
6.34	Welfare distributions with 20% MOT whitewashers	112
6.35	Welfare distributions for different professions (MCS) (1)	114
6.36	Welfare distributions for different professions (MCS) (2)	114

6.37	Growth of median welfare (MCS)	114
6.38	Welfare distributions for different professions (<i>i</i> -WAT) (1)	115
6.39	Welfare distributions for different professions (<i>i</i> -WAT) (2)	115
6.40	Welfare distributions for different professions (NEO) (1)	116
6.41	Welfare distributions for different professions (NEO) (2)	116
6.42	Growth of median welfare (NEO)	117
6.43	Welfare distributions for different professions (NEO') (1)	118
6.44	Welfare distributions for different professions (NEO') (2)	118
6.45	Welfare distributions for different professions (MCS ⁻) (1)	119
6.46	Welfare distributions for different professions (MCS ⁻) (2)	119
6.47	Example: redeemed ticket (1)	122
6.48	Example: redeemed ticket (2)	122
6.49	Example: one of tickets with the longest chain	123
6.50	Typical public key ring	124
6.51	Resulted signature network (web of trust)	124
6.52	Resulted trust network (barter relation)	125
6.53	Resulted validity network (validation relation)	125
6.54	Distribution of web of trust (1)	126
6.55	Distribution of web of trust (2)	127
6.56	Histogram of drawn/used tickets	127
6.57	Histogram of received tickets	128
6.58	Accumulated number of trades over time	129
6.59	Age-distribution of participants	129
6.60	Gender-distribution of participants	130
6.61	Frequencies of trade types	131
6.62	Age-distribution of ticket drawers/users	131
6.63	Age-distribution of passive or non-users	132
6.64	Questionnaire to the participants	134
6.65	Did you know about complementary currencies?	135
6.66	How much did you understand about <i>i</i> -WAT?	136
6.67	How much do you anticipate from <i>i</i> -WAT?	137
6.68	Do you think complementary currencies will be useful?	138
6.69	Successful downloads of <i>wija</i> May-June 2005	139
6.70	Successful downloads of <i>wija</i> November-December 2005	140
6.71	Successful downloads of <i>wija</i> version 0.11 (~ Jan/07/2006)	141
6.72	Mean time to approval and strength of presence	143
6.73	Examples of purchased goods	144
6.74	Example: issued <i>reduction</i> ticket	145
6.75	Example: <i>barter YEN</i> in circulation	146
6.76	The design of <i>barter Dollar</i>	146
7.1	<i>i</i> -WAT and Geek Credit/Ripple trust models	156
7.2	<i>i</i> -WAT and Karma trust models	157

- 9.1 A model of WAT/*i*-WAT for post-catastrophic recovery . . . 167
- B.1 An example of the content of a data description file 185

List of Tables

2.1	Classification of complementary currencies	10
2.2	Terminology in P2P	13
2.3	Classification of P2P currencies	14
3.1	<i>i</i> -WAT messages	26
4.1	Roles of layers in the five-layer model	35
4.2	Possible misbehaviors and the imposed risks to the subjects	54
4.3	Evasive and graceful actions	55
4.4	Meanings of tickets in the <i>i</i> -WAT book of a user	59
5.1	Built-in plug-ins for <i>wija</i>	70
5.2	JEPs implemented in <i>wija</i> and its bundled plug-ins	71
5.3	Chronological list of experiments	77
5.4	Chronological list of <i>wija</i> releases	79
5.5	Initial properties of the small worlds	83
5.6	Common parameters for the simulations	85
6.1	Product types and production/consumption rates	113
6.2	WIDE Hours statistics from Sept. 2003	120
6.3	WIDE Hours statistics from Mar. 2004	120
6.4	MANA statistics October 26, 2003 ~ February 28, 2004	121
6.5	Properties of the resulted web of trust	125
6.6	Replies to the questionnaire (ordinal-scale data only)	133
6.7	Statistical data from 4 participants (April 2005)	142
6.8	Statistical data from participant <i>A</i> (Apr 2005, Jan 2006)	142
6.9	Participants and their approval behaviors (April 2005)	143

Nomenclature

BSSID	Basic Service Set Identifier (for a wireless network).
CCS	Content Cruising System [41].
DAMD	Distributed Algorithmic Mechanism Design [21].
DHT	Distributed Hash Table [40].
GNU	GNU's Not UNIX [25].
GnuPG	GNU Privacy Guard [93]. An implementation of OpenPGP.
gnuplot	A plotter software [110].
GPL	GNU General Public License [24].
GUI	Graphical User Interface.
HTTP	Hypertext Transfer Protocol [7, 23].
IDEON	Integrated Distributed Environment with Overlay Network [31]. A working group at WIDE Project.
IETF	Internet Engineering Task Force [95].
IP	Internet Protocol [63, 19].
Ithaca HOURS	A complementary currency in the region of Ithaca, New York [29].
iTunes	A music player software by Apple Computer, Inc. [3].
<i>i</i> -WAT	The Internet WAT System [74].
J2ME	Java 2 Micro Edition [91].
J2SE	Java 2 Standard Edition [91].
J9	A lightweight J2ME implementation by IBM Software [38].
Jabber	An open system for instant messaging and presence sharing [42].
JAR	Java ARchive. A compression format for Java executables.
JEP	Jabber Enhancement Proposal [42].
JNI	Java Native Interface.
JSF	Jabber Software Foundation [42].
LETS	Local Exchange Trading System [87].
Linux	Also known as GNU/Linux. A free Unix-type operating system originally created by Linus Torvalds [47].

Mac OS X	A Unix-type operating system by Apple Computer, Inc. [4].
MANA	A web-based MCS[59].
MCS	Mutual Credit System [84].
MOT	Multiplication Over Time [79].
MTTA	Mean Time To Approval.
NEO	New Economic Order.
OMELETS	Open, Modular and Extensible LETS [75].
OpenPGP	An open standard for PGP [11].
P2P	Peer-to-Peer.
Pajek	A software for large network analysis [54].
PGP	Pretty Good Privacy.
Pocket GnuPG	A version of GnuPG ported to Pocket PC.
Pocket PC	A handheld computer to run a version of Windows CE operating system [105].
QuickTime	A family of digital media creation, delivery and playback software by Apple Computer, Inc. [2].
R	Also known as GNU S. A software for statistical computing and graphics [96].
RFID	Radio Frequency IDentification.
ROT	Reduction Over Time [80, 81].
SHA-1	Secure Hash Algorithm 1 [60].
SIQR	Semi Inter-Quartile Range.
SMTP	Simple Mail Transfer Protocol [64].
SNS	Social Networking Service.
SOCKS5	SOCKeTS protocol version 5 [45].
Swing	The standard GUI toolkit on J2SE [91].
SWT	Standard Widget Toolkit for Java provided by Eclipse [92].
WAT System	An autonomous distributed barter currency in real life [99].
WIDE Project	A project toward Widely Integrated Distributed Environment [101].
WIDE Hours	A barter currency for WIDE members [75].
Windows	A family of operating systems by Microsoft Corporation [50].
XML	Extensible Markup Language [9].
XMPP	Extensible Messaging and Presence Protocol [69, 70].

Chapter 1

Introduction

The economics of the future are somewhat different. You see, money doesn't exist in the 24th century. . . . The acquisition of wealth is no longer the driving force in our lives. We work to better ourselves and the rest of humanity.

*– Capt. Jean-Luc Picard
(from “Star Trek – First Contact”)*

1.1 Economics in the Presence of Replicators

1.1.1 Economics of the Star Trek Universe

The money-less economics of the 24th century must owe its existence to the invention of *replicators*, which is one of the core technologies in the fictional universe of Star Trek. A *replicator* is a machine capable of converting energy into matters and vice versa[106]. It can replicate food and water onboard starships out of abundant subatomic particles in the universe.

One can easily imagine that this technology must have helped humanity getting rid of poverty for good, and brought equal welfare to all; since *replicators* themselves should easily be replicated for no price, there is no point in setting prices for the machines and their products, and anyone should be able to use it to satisfy their physical needs for food and water to continue their lives, and garments and housings to keep them warm.

1.1.2 Motivation for This Study

What motivates this study is the presence of another kind of *replicators*, which do not replicate matters but information, which we can readily use today. They are called *network of digital computers* connected one another via the Internet. The presence of the global digital network has reduced

the cost of copying and distributing information to an extremely low level, which can potentially bring inevitable economic impacts to our lives.

Already, there are venders of information to whom money seems to be attracted, which is a scarce medium to begin with. This may result in a harsher division of haves and have-nots in our world, which the author wishes to prevent from happening.

There is a need for designing a new society, accommodating the impacts the Internet is bringing to communication, logistics and ways of our lives in general.

The author believes that money should not be involved in exchanging information on the Internet, or at least, we should not depend so much on money when we make exchanges. Still, having exchange media will help as the author will illustrate later, but they should be something more like the ones presumably used in the Star Trek universe, different from the present form of money. The author hopes that through this dissertation, we together will discover what such media would possibly look like and how they can be used.

For a start, the author believes that such media should be more autonomous and distributed than the conventional money is, having no fixed authority so that they can bring equal opportunities for exchanges and welfare to all; the media will show characteristics of peer-to-peer (P2P) as described in section 2.2.

1.2 Dilemmas

There are some dilemmas in concern when we think about economics of a distributed system consisting of autonomous peers. Having these in mind is important because the presence of exchange media can be both a solution and a cause of such dilemmas.

1.2.1 Asymmetry in the Possible Outcomes

Exchanging is a necessary building block of P2P systems, which can potentially harness the under-utilized power of the network of computers connected one another via the Internet. Such power is often used for sharing values in real life, such as music, video, or even physical goods such as furnitures by way of auctions.

There seem to be three classes of values or resources that can be made subjects of exchanges in P2P systems:

1. Atoms, or physical goods, that can be stockpiled for future use,
2. Bits, or information, that can be digitally copied, and
3. Presences, or labors, that cannot be digitally copied or stockpiled.

(Time slots for computing resources are typical presence-type values in P2P systems.)

If these are equally treated as commodities, however, the economy of the system is likely to collapse, because information can be reproduced at a negligibly small cost whereas labors cannot even be stockpiled for future use. This will result in an especially advantageous situation for bit providers, and disadvantageous situation for presence providers.

Meanwhile, licenses such as GNU GPL[24] or those produced by Creative Commons[17] allow certain types of digital information to be freely shared. Although this is an emerging movement, this would force producers of information to participate in exchange systems as ones with low productivities. This will then result in a disadvantageous situation for bit providers.

In either case, presence of information, highly reproducible resource, is likely to cause asymmetry in the outcomes of exchange systems based on existing media. The author believes that this is a dilemma to be solved.

1.2.2 Tragedy of the Commons

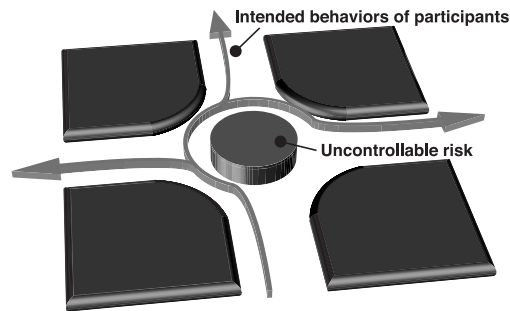
G. Hardin introduced the concept of the tragedy of the commons in [34].

Suppose there is an open pasture, and each herdsman rationally seeks to maximize their own benefits. The utility to him or her of adding one more animal to their herds has a negative and positive component. The positive component is the profit to be expected from the extra animal, which the herdsman can solely take. The negative component is the cost of additional overgrazing created by the extra animal, but the cost is shared by all the herdsmen. Therefore every herdsman concludes that the only sensible course of action is to add an animal to their herds. The situation is not changed by this addition, so that everyone adds another, and another – they will keep adding animals until all grass is gone. Everyone seems to be locked into a system that compels them to increase the sizes of their herds with no limit, in a world which has a limit.

This is a problem of *free-riding* bringing ruin to all. It is a dilemma in distributed systems consisting of self-interested peers, because pursuit of self-interest turns out to be harmful to the welfare of their own.

1.2.3 Risks and Moral Hazards

Moral hazard is another dilemma in the design of distributed systems with autonomous participants: an incentive is created by protections against risks that influences the participants to undertake greater risks than when they were not protected, because any consequences would be taken care of by the protections. This hazard is present whether the participants take advantage of the rules intentionally or inadvertently.



* If participants deviate from the intended behaviors, they must confront the uncontrollable risk.

Figure 1.1: A roundabout and its incentive mechanism

For example, traffic signals and stop signs at intersections are intended to protect drivers from accidents, but they turn out to be sources of moral hazards; those protections allow drivers to be less careful about the speed at which they are driving their vehicles, resulting in accidents when their attentions fail.

One successful solution to this problem is a *roundabout*, or traffic circle, which is a circular one-way road with an obstacle at the center (Figure 1.1). A study[66] showed that converting traffic signals and stop signs to roundabouts would result in crash severities at intersections reduced by 38%. This is attributed in large part to the fact that, at a roundabout, drivers *must* slow down their vehicles *spontaneously*; otherwise there is an uncontrollable risk of crashing into the obstacle.

This example demonstrates the effectiveness of incentive mechanisms as counteractions against misbehaviors out of moral hazards. When applying such mechanisms to decentralized systems, however, the uncontrollable risks as disincentives may not come from central authorities; they must be imposed by the behaviors of other participants.

1.3 Thesis Statement

This research is to propose a distributed autonomous barter currency which facilitates exchanges of information as well as physical resources, whose safety is maintained by rational behaviors of participants themselves. The currency is intended to be used in computer network systems, especially those showing peer-to-peer (P2P) characteristics (see section 2.2), as well as in real life. This research tries to give solutions to problems on collaboration among selfish peers on the Internet who build mechanisms for autonomous,

distributed cooperative activities, by providing a platform for exchanges.

The author proposed *i*-WAT[74] in year 2003 as a currency usable on the Internet based on the WAT System[99], a polycentric barter currency using *WAT tickets* as its media of exchange. A WAT ticket is like a bill of exchange, but without a specified redemption date or place. *i*-WAT implements the tickets electronically by exchanging messages signed in OpenPGP[11]. It has been in practical use since June 2004.

This dissertation begins by describing the core designs of WAT/*i*-WAT and the trust and incentive models of *i*-WAT. It then shows that the design of *i*-WAT is incentive-compatible[21] as to protections against strategies and counteractions against moral hazards: taking advantage of the rules will result in the subject's confrontation to an uncontrollable risk. Since *i*-WAT has no fixed authority, such risks are imposed by rational behaviors of other participants. This is by no means denial of the threats, but the author shows that it provides a reasonable level of security by a series of simulations.

The strategies and hazards in concern will include whitewashing (or repetitive re-entrance with new identities), impostors, unintentional breach of trust and collusions. Those threats which take advantage of vulnerabilities of implementations of the system instead of its trust/incentive models, such as *denial of service*, are out of scope of this research.

1.4 Terminology

In this section, the author gives definitions to fundamental terms necessary to continue this dissertation.

1.4.1 Exchange-Related Concepts

Definition 1 (barter) *Barter is to trade goods or services without the exchange of money (definition by the American Heritage). In this dissertation, the meaning of the word is extended to include "reciprocal trades" described in [49]:*

Currently there are numerous "barter clubs" in operation. Strictly speaking, they are generally not barter clubs in that they utilize some "barter currency" as medium of exchange. Some call this type of trade "reciprocal trade." It is characterized by utilizing a medium of exchange other than those issued by coercive political agencies.

Figure 1.2 illustrates the concept of barter economy in this dissertation.

If a barter is to be strictly performed without any exchange medium (on the left of the figure), it requires a double coincidence of wants: Alice (A) needs to want what Bob (B) can supply, and Bob needs to want what Alice

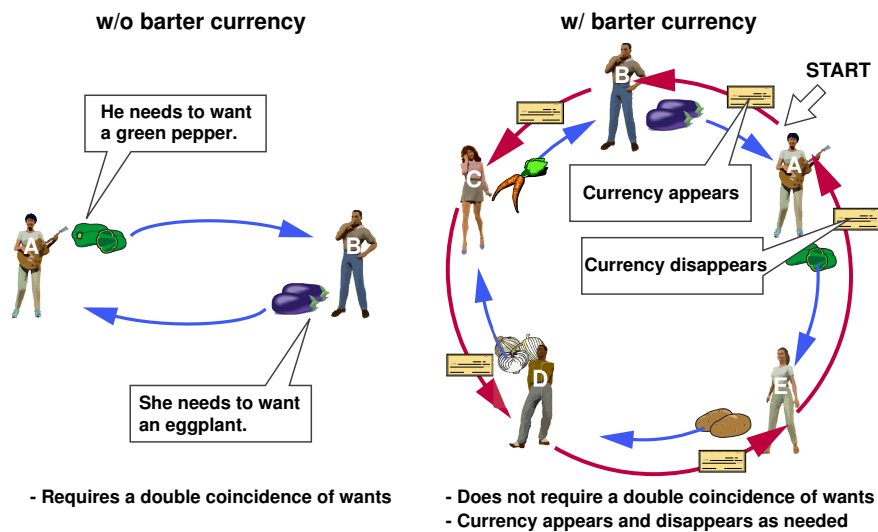


Figure 1.2: A barter with or without currency

can supply. It is not likely that such a double coincidence actualizes often; for example, Alice may want an eggplant, but Bob may want a carrot which Alice cannot supply. There is only a limited chance that an exchange will ever take place.

If, on the other hand, a barter can be performed with an exchange medium (on the right of the figure), not only it eliminates the necessity for a double coincidence of wants, but it may also facilitate more barter to take place; let the author explain with the same example.

Suppose that in return for an eggplant, Alice generates a medium of exchange, and gives it to Bob. Bob then gives it to Cameron (C) in exchange for a carrot. The medium of exchange circulates among people until it is finally given back to Alice in exchange for a green pepper, at which time the medium ceases to exist. A barter currency appears and disappears as situations require it. It is, in the end, eliminated from the system as if it did not exist at all, making the final state indistinguishable from the result of a complex multi-barter arrangement. This is how a barter currency is different from the conventional form of money.

In this dissertation, the word *currency* is used primarily to denote such barter currencies.

Definition 2 (currency) *Currency is money in any form when in actual use as a medium of exchange (definition by the American Heritage). In this dissertation, the meaning of the word is extended to include barter currencies.*

As explained above, the currency to be proposed needs to have important properties with respect to its appearance and disappearance.

Property 1 (on-demand appearance) *One should be able to generate a medium of exchange when they need it.*

Property 2 (off-demand disappearance) *A medium of exchange must disappear once its existence is no longer required.*

1.4.2 Safety-Related Concepts

Safety in a distributed system is that something bad does not happen[83], so that the invariants of the system are maintained during its execution. Strategies and moral hazards are among threats to the safety of a system consisting of autonomous participants.

Definition 3 (strategy-resistance) *A strategy is a dishonest behavior such that the gain of the player is unfairly increased. A system is resistant to strategies if its incentive mechanism is both strategyproof and group-strategyproof.*

Strategyproof and *group-strategyproof* are well-known concepts in the field of mechanism design[21].

Property 3 (strategyproof) *A mechanism is strategyproof if no participant has an incentive to lie.*

Property 4 (group-strategyproof) *A mechanism is group-strategyproof if at least one participant in a group is necessarily victimized by collusion among the group members.*

In this dissertation, strategies and moral hazards are distinctly treated although they are related concepts.

Definition 4 (moral-hazard resistance) *A system is resistant to moral hazards if a disincentive is created by protections against risks that influences the participants not to undertake any greater risks.*

Chapter 2

Background

2.1 Historical Perspective

The author believes that there are many important lessons we can learn from the past, real-life experimental currencies, with which we can improve the design of the currency to be proposed.

2.1.1 Complementary Currencies

Money is a well-known medium of exchange, but its scarcity has caused a lot of problems: it is something we compete for, it necessarily creates haves and have-nots, and economy is slow where it is in short supply. *Complementary currencies*, or alternative forms of monetary media, have been proposed and tested in real life to achieve an autonomous, sustainable local economy even in short of money. There have been successful cases, such as experiments in Wörgl in 1932 (stamp scrip[86]), in Comox Valley in 1983 (LETS: Local Exchange Trading System[87]) and in Ithaca since 1991 (Ithaca HOURS[29]).

Those currencies are either apparent cases of barter currencies (stamp scrip and LETS among the above examples) or transformable into apparent barter currencies without changing their semantics (Ithaca HOURS).

Being generated closer to the places in need, those currencies are used to support values which are not readily circulated in today's economy, such as volunteer works, daily helps and enjoyments, or skills that are not regularly utilized. Studies of such currencies would benefit the designs of P2P systems, which are also intended to make use of under-utilized resources.

2.1.2 Classification

Complementary currencies can be classified into categories shown in Table 2.1.

Definition 5 (debt-oriented currency) *A currency is debt-oriented if*

Table 2.1: Classification of complementary currencies

	<i>Centralized</i>	<i>Not centralized</i>
<i>Debt-oriented</i>	MCS (LETS, Ithaca HOURS)	The WAT System
<i>Labor-oriented</i>	Stamp scrip, Time Dollars[97]	–

creation of an exchange medium implies that someone in the system is in debt.

Free-riding in a debt-oriented currency is when that someone manages without repaying the debt. It is so identified only when that someone leaves the system, because as long as they are in the system, their debt is in the record. *Whitewashers* free-ride repetitively, dealing with which we can reason about free-riding while maintaining the constant population in the system.

Definition 6 (labor-oriented currency) *A currency is labor-oriented if creation of an exchange medium is limited to the case where a labor is performed by someone in the system.*

It can be viewed that, in a labor-oriented currency, the system itself becomes in debt, instead of someone in it, upon creation of an exchange medium.

MCS: Mutual Credit System

Many of the existing currencies fall into the category of MCS[85] (Mutual Credit System), a form of debt-oriented currency.

Concept An MCS is an accounting system of exchange, in which there is no initial stock of cash. When a member joins, the initial balance of their account is set to zero. The system allows negative balance to some predefined extent, and values are transferred between accounts by subtracting an amount at one end and adding the same amount at the other, as two members participate in a trade.

Implementations LETS is the most common implementation of MCS which does exactly the above.

Many other currency systems can be viewed as different forms of MCS by appropriate transformations without changing their semantics. Ithaca HOURS, for example, issues paper notes of the unit “HOURS” when someone joins the system, which can be circulated within the region. This is transformed into an MCS by setting up a virtual account for each person involved in circulation of the notes, subtracting the amount which is worth

the HOURS from the balance of the first person, and then transferring the amount to the accounts of those who are involved one by one.

Safety and Risks Theoretically, all accounts sum to zero in an MCS, which can be defined as a safety of the system. However, maintaining this in practice is problematic.

If one has a deficit on one's account in an MCS, the debt is owed to the rest of the members[28]. Therefore everyone shares the same averaged level of risks that the debt remains unpaid, which may allow participants to be indifferent to the presence of the risks, thus tempting them toward moral hazards.

Sometimes, the expectation of the loss is compensated in the forms of maintenance fees, deposits or demurrages.

The WAT System

The WAT System is a decentralized form of debt-oriented currency which will be described in detail in section 3.2, on which the currency to be proposed is based. It has been used in practice in some regions in Japan: yufu[112] and CHITA WAT[14] are some of the examples.

2.1.3 Reduction Over Time

It is known among the practitioners of complementary currencies that reducing the value of the exchange medium over time accelerates spending. The experiment in Wörgl in 1932 is a well-known example. It was based on the idea of *stamp scrip* (Figure 2.1) introduced by Sylvio Gesell in [27], who believed exchange media must also deteriorate as the exchanged goods do (for this reason, *Reduction Over Time* is dubbed *ROT* in this research).



Figure 2.1: A sample of stamp scrip used in Mason City, Iowa, in 1930's

A user of stamp scrips needs to paste a stamp every week on the back of a scrip, or the scrip becomes invalid. Users are motivated to spend the scrip before another stamp is required. The stamp is like a tax for withholding exchange media; this seems to be a suitable way to implement ROT in a centralized currency system.

If ROT is to be implemented for decentralized currencies, a different approach needs to be taken. In [80, 81], the author has applied the notion of *calendar money* (Figure 2.2) by Arthur Dahlberg[18], which has a schedule of reduction printed on the note, to achieve it in the proposed currency system; the author has proposed the introduction of *reduction tickets* (section 4.5) whose values are reduced over time.



Figure 2.2: A sample of calendar money in Salt Lake City, Utah, in 1930's

2.1.4 Multiplication Over Time

But varying over time needs not to be limited in one direction. There is another example of a real-life complementary currency, called MAAS[48], whose exchange medium increases its value over time. A MAAS ticket is intended to be issued by an artist, who promises to provide (artistic) goods in the future which will worth more than the value being exchanged in an ongoing trade. MAAS helps those who want to create something but lack resources to do so at the moment.

Likewise, *Multiplication Over Time* can be used in peer-to-peer systems to facilitate exchanges of data or services which are strongly needed by some parties to create new data or services (new values), without sacrificing their current ownerships of resources. They can even control the timing of redemption to some extent by an incentive mechanism.

Applications may include collection of sensory data from many locations to provide value-added services, such as weather or traffic forecasting¹.

¹ This may look contradicting to the later assertion that information should not be vended as a commodity (section 4.12). Perhaps one way to conceive such a system is to think about exchanges in terms of storage, bandwidth and CPU time required for storing, transferring and processing the data involved.

The author has proposed the introduction of *multiplication* tickets (section 4.6) to the proposed currency system, whose values are multiplied over time.

2.2 Peer-to-Peer Network

2.2.1 Concept

Definition 7 (peer) *A peer is a participant in a system, human or otherwise, who shares the same set of roles as others.*

Definition 8 (peer-to-peer) *A P2P network is a group of peers which form an overlay network as an infrastructure to realize free and creative rendezvous, location and routing.*

Table 2.2 shows the terminology to be used in this dissertation with respect to P2P, compiled through discussions among members of IDEON[31], a group of researchers and practitioners in the field.

Table 2.2: Terminology in P2P

<i>Terms</i>	<i>Meaning</i>
Free	Having no restriction whatsoever as to with which peers one can communicate.
Creative	Being able to select a set of peers according to one's objectives, requirements, needs and contexts so that communication becomes most valuable for the participants.
Rendezvous	To identify such a peer.
Location	To locate such a peer in the overlay networks by the acquired identifier.
Routing	To deliver a message to such a peer on the acquired location.
Overlay Network	An application-specific virtual network of peers over the IP network to realize rendezvous, location and routing over an appropriate abstraction of entities.

In comparison with more centralized or asymmetrical models, P2P excels in ability to self-organize and plasticity; participants can spontaneously start and sustain such systems even in the presence of unrecoverable partial failures.

As [46] suggests, P2P has both technical and social components. From a technical perspective, P2P can potentially harness the under-utilized computing power, storage space or input/output capabilities of the network of digital computers connected one another via the Internet so that such excessive abilities can be utilized for the peers who need them. From a social perspective, P2P implies new person-to-person interaction structures or new organizations that can empower people by releasing the under-utilized resources of individuals to the communities.

2.2.2 Peer-to-Peer Barter Currencies

Exchanging is a necessary building block of P2P systems. Since the resources are distributed over autonomous entities, such exchanging needs to be performed in an *incentive-compatible*[21] way: the coordination must be accomplished by collection of selfish behaviors. A medium of exchange which represents a guaranteed value should take an important role in the designs of P2P systems. Such medium of exchange also needs to be P2P (showing such characteristics as autonomy and decentralization), or otherwise, desired properties such as ability to self-organize or plasticity will be lost.

Many such media have been proposed, many of which take a form of barter currency. They can be thought analogous to the existing complementary currencies as shown in Table 2.3.

Table 2.3: Classification of P2P currencies

	<i>Centralized</i>	<i>Not centralized</i>
<i>Debt-oriented</i>	MCS (MojoNation, Karma), Multiple MCS (Ripple)	<i>i</i> -WAT, Geek Credit, PPay, Samsara
<i>Labor-oriented</i>	–	BitTorrent

MojoNation[6], Karma[98] and Ripple[26] are examples of MCS-counterparts in the P2P context. Geek Credit[44] and PPay[111] resemble the WAT System, as well as *i*-WAT does, which the author has proposed, implemented and been deploying.

It is debatable whether Samsara[16] and BitTorrent[15] are systems of currencies or not, because they facilitate exchanges by the subjects of exchanges themselves such as storage space and bandwidth, respectively, but they certainly are systems of exchange. It is also debatable whether or not a decentralized labor-oriented currency is a possible concept, because if some peer performs a labor for another, then it implies that the second peer is in debt to the first peer. Which makes the system indistinguishable from being debt-oriented.

Exchanges in P2P systems can be naturally facilitated by debt-oriented currencies, because utilizing computing and communication resources of other computers is a debt to the owners of such resources, which can be expressed by a medium of exchange that represents a debt.

2.3 Relevant Theories

2.3.1 Distributed Algorithmic Mechanism Design

Distributed Algorithmic Mechanism Design (DAMD)[21] combines computational tractability with incentive compatibility and distributed computing.

DAMD, to the best knowledge of the author, is the only available theoretical framework which addresses all necessary ingredients with respect to incentive mechanisms in the designs of Internet-centric, autonomous distributed collaborative systems.

Figure 2.3 illustrates the model of mechanism design. A mechanism is an

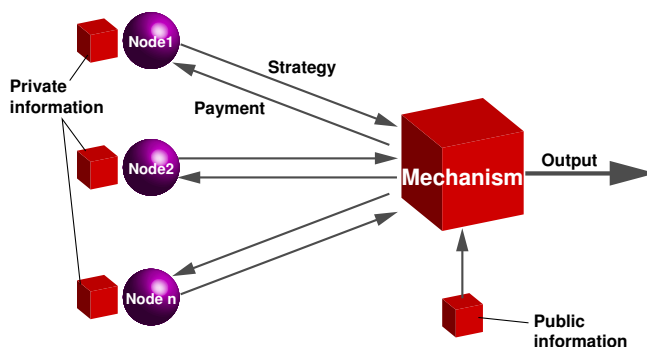


Figure 2.3: The model of mechanism design

output specification and payments to agents which motivate them to behave in ways which will lead to the desired system-wide goal. [21] explains an application of DAMD as follows:

For example, consider the problem of routing. Agents may be individual routers within a network or entire autonomous domains. Each agent incurs a cost when it transports a packet, and this cost is known only to the agent, not to the mechanism designer or to the routing protocol. Each agent is required by the protocol to declare a cost. The system-wide goal is to have the routing protocol choose the true lowest-cost path between any two agents in the network.

DAMD is relevant to this research by all means, and the author will borrow from its concepts from time to time. But the author sees that the framework is too generic, which necessitates solutions to be too specific or dependent on specific problems, because definitions of incentive compatibility and computational tractability depend on the particular problems.

The author is to construct a more concrete framework around a currency, so that the theory will become specific while the solutions can be more generic and applicable to many problems.

2.3.2 Digital Signature

Digital signature is an essential technology for designing a dependable economic medium, which can provide a proof of debts or credits.

Throughout this dissertation, the author uses notations from [10] for formalization, with additional abstractions built upon them to fit our purposes.

Suppose Alice (A) is associated with a public/secret key pair denoted as $\langle K_A, K_A^{-1} \rangle$. To simplify the arguments to follow, we assume that each user has exactly one key pair associated with them.

A digital signature has two objectives:

1. To prove that Alice once admitted a message m .
2. To prove that m has not been altered since then.

These can be realized by encrypting m with Alice's secret key K_A^{-1} , obtaining $\{m\}_{K_A^{-1}}$ which is only decrypted with her public key K_A . Since K_A^{-1} is a secret known only to Alice, those who could decrypt $\{m\}_{K_A^{-1}}$ can infer that it must have been encrypted by Alice. They can also be certain that m has not been altered since Alice made $\{m\}_{K_A^{-1}}$ if the result of decryption equals m .

Usually, for efficiency reasons, instead of encrypting m itself, a digital signature is made by applying a secure hash function H to m , then encrypting the hash value with the secret key. H must be carefully chosen so that it is computationally infeasible to obtain m' where $m' \neq m$ such that $H(m) = H(m')$.

Definition 9 (digital signature) We write $A \xrightarrow{\text{signs}} m$ if and only if A presents both a plain-text message m and its encrypted form $\{H(m)\}_{K_A^{-1}}$. The latter is called a signature on the former.

The signature can be verified by Bob if he has a copy of Alice's public key K_A . To verify the signature, he calculates $H(m)$ from m , decrypts $\{H(m)\}_{K_A^{-1}}$ with K_A , and compares the two resulted values.

One question is how Bob can be sure that his copy of Alice's public key is genuine.

Definition 10 (validating relation) $x \xrightarrow{v} y$ if x possesses a copy of y 's public key K_y , and infers that the copy is genuine.

We also write $x \overset{v}{\leftrightarrow} y$ if and only if $x \xrightarrow{v} y \wedge y \xrightarrow{v} x$ (mutually validating relation).

A trust model around validity of public keys is a specific definition of *validating relation* \xrightarrow{v} in the system in concern. Typically, validity of a public key is supported by a *certificate*, or a signature on the key. For example, if

Bob (B) sees Cameron (C) such that $B \xrightarrow{v} C \wedge C \xrightarrow{\text{signs}} K_A$, then $B \xrightarrow{v} A$ assuming that C 's certificate is trustworthy. This relation is recursive, so that someone needs to self-certify at some point.

A public key infrastructure uses a tree of *certificate authorities*, or issuers of certificates, whose public keys are validated by the parent nodes, rooted by a self-certifying authority.

2.3.3 Web of Trust

In a *web of trust*, however, responsibility for validating public keys is delegated to people one trusts, without necessitating certificate authorities. It is a network of people signing the public keys of others.

Signing relation \xrightarrow{s} states that one certifies that its copy of someone's public key is genuine.

Definition 11 (signing relation) \xrightarrow{s} is defined as follows:

1. $x \xrightarrow{s} x$
2. $x \xrightarrow{s} y$ if $x \xrightarrow{\text{signs}} K_y$

We also write $x \xleftrightarrow{s} y$ if and only if $x \xrightarrow{s} y \wedge y \xrightarrow{s} x$ (mutually signing relation).

Definition 12 (signing-apart relation) $\xrightarrow{s[n]}$ is defined as follows:

1. $x \xrightarrow{s[0]} x$
2. $x \xrightarrow{s[1]} y$ if $x \xrightarrow{s} y \wedge x \neq y$.
3. $x \xrightarrow{s[a+b]} z$ if $x \xrightarrow{s[a]} y \wedge y \xrightarrow{s[b]} z$.

We also write $A \xrightarrow{s} B \xrightarrow{s[n]} C$ in place of $A \xrightarrow{s[n+1]} C$ if $A \xrightarrow{s} B \wedge B \xrightarrow{s[n]} C$ (expansion of signing-apart relation) in order to clarify who stands in between the chain of signing relations.

Definition 13 (web of trust) A *web of trust* for x is a set of all y such that $x \xrightarrow{s[n]} y$ where $n \geq 0$.

A specific validation relation needs to be defined over a web of trust. PGP (Pretty Good Privacy) is an example of a cryptographic technology which defines such a relation. We use GnuPG[93] as our choice of implementation of OpenPGP[11] standard.

2.3.4 PGP Trust Model

Let us further define that \mathcal{T}_x is the set of users x considers fully trustable, and \mathcal{T}'_x is the set of users x considers marginally trustable.

In the context of PGP public key signing, *fully trustable* means that one considers that the owner of a public key has an excellent understanding of key signing, and his or her signature on a key would be as good as their own, and *marginally trustable* means that one considers that the owner of a public key understands the implications of key signing and properly validates keys before signing them[94].

The PGP trust model is a definition of *validating relation* \xrightarrow{v} over a web of trust.

Definition 14 (PGP trust model) $x \xrightarrow{v} y$ if

1. *sufficient number of valid key owners sign y 's public key, i.e.*
 - (a) $x \xrightarrow{s} y$, or
 - (b) *there exists at least f instances of z such that $z \in \mathcal{T}_x$, $x \xrightarrow{v} z \wedge z \xrightarrow{s} y$, or*
 - (c) *there exists at least m instances of z such that $z \in \mathcal{T}'_x$, $x \xrightarrow{v} z \wedge z \xrightarrow{s} y$; and*
2. $x \xrightarrow{s[n]} y$ where $n \leq h$,

where f , m and h are the required number of fully trusted key owners, required number of marginally trusted key owners, and number of maximum steps in the path in the web of trust tracing x back from y , respectively.

We define the marginally validating relation (\xrightarrow{v}) as follows:

Definition 15 (weak PGP trust model) $x(\xrightarrow{v})y$ if

1. *insufficient number of valid key owners sign y 's public key, i.e.*
 - (a) *there exists at least one but less than f instances of z such that $z \in \mathcal{T}_x$, $x \xrightarrow{v} z \wedge z \xrightarrow{s} y$, or*
 - (b) *there exists at least one but less than m instances of z such that $z \in \mathcal{T}'_x$, $x \xrightarrow{v} z \wedge z \xrightarrow{s} y$; and*
2. $x \xrightarrow{s[n]} y$ where $n \leq h$

By default, GnuPG defines $f = 1$, $m = 3$ and $h = 5$.

Chapter 3

Problem Statements

3.1 Requirements

Requirements for the P2P barter currency to be proposed are autonomy and safety, as well as integration such that it can be used as the platform for various transactions and activities.

3.1.1 Autonomy

First, the author clarifies the level of autonomy required for the proposed currency, or decides the extent to which it must be able to bring equal opportunities for exchanges. To do so, the author argues that we should assume extreme cases of recessions, social instabilities, natural disasters and unavailability of technologies (except for the devices to run the software).

We cannot assume the existences of functioning governments or administrations of any form. We cannot assume the presence of a network infrastructure if we consider activities in places not IP-reachable or after catastrophic events.

Required autonomy is categorized into the following:

1. Administration-freedom

It must be free from necessity for administrative entities in achieving *on-demand appearance* (Property 1) and *off-demand disappearance* (Property 2) of the medium of exchange.

2. Interference-freedom

It must be operable independently from outside economy. Operation of the currency must be effectively free in terms of legal tenders, and the currency should not be linked to the legal tenders; the system must be self-sufficient in terms of its economy.

3. Locality-freedom

It must be ubiquitously operable. Not only it implies its need to be operable over the Internet, it must at the same time be free from necessity for Internet-connectivity, in order to be usable even in cases of post-catastrophic situations. This implies its need to be operable over (wireless) ad-hoc networks in addition to the Internet.

3.1.2 Safety

The author clarifies the set of safeties whose necessities are imposed by the level of autonomy explained above.

Required safeties are categorized into the following:

1. Idempotence

One of safety-related properties of a distributed system is that delay of a message and its omission are not essentially distinguishable. Therefore, a receiver may receive the same message multiple times, which has been resent by the sender. *Idempotence* is a property which states that such multiple receipts of the same message have no effect on the correctness of the system.

In particular, in a currency system, the problem of double-spending, or multiple use of the digital copies of the same identical exchange medium, is closely related with the concept of idempotence, as it involves multiple receipts of the same message.

2. Fault-tolerance

The currency system must try to mask faults where it is possible; it must restore itself from a faulty state within a reasonable time period.

3. Plasticity

If a fault cannot be masked, it must be able to detach the faulty part from the system, and maintain the soundness of the remaining part.

4. Privacy

It must not reveal private information of participants to others where it is unnecessary. Such information may include the following:

- (a) Identities (such as e-mail addresses and PGP photo IDs)
- (b) The content of trading
- (c) Traffic (the trade relations among participants)

There is a tradeoff between privacy and other safeties. Strategy-resistance, for example, may require the participants to publish some information which would be considered private otherwise.

5. Strategy-resistance

It must show the characteristic as defined by Definition 3.

6. Moral-hazard resistance

It must show the characteristic as defined by Definition 4.

3.1.3 Integration

The author clarifies a series of characteristics required for the currency to be a working system, and sustain to be so for the future.

Integration should achieve the following as requirements:

1. Openness

It must be open so that anyone can contribute computing resources to sustain its existence. It must welcome anyone to contribute software to improve availability of the system and its adaptability to new hardware and operating system platforms.

2. Extensibility

It must be extensible by anyone who has a need to do so. In particular, anyone must be able to design specific currencies out of this system to satisfy their needs.

3. Coexistence

It must be able to coexist with other currency systems for deployment reasons.

4. Activity-coordination

It must provide facilities for collaboration so that the currency will become more usable. In particular, it must provide facilities so that participants can satisfy preconditions for trades, such as rendezvous, public-key exchanges and validations.

3.2 The WAT System

The WAT System is the basic currency model the author has chosen, upon which the proposed P2P barter currency is designed.

3.2.1 Overview

The WAT System[99] is a barter currency designed by Eiichi Morino, who has been giving advices to this research.

A *WAT ticket*, a sheet of paper resembling a bill of exchange, is used as the medium of exchange in the system. A lifecycle of a WAT ticket involves three stages of trading as illustrated in Figure 3.1:

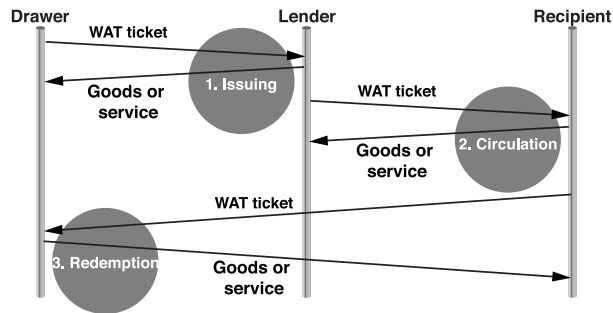


Figure 3.1: Three stages of trading with a WAT ticket

Definition 16 (the WAT Core) *The WAT Core is the protocol of issuing, circulation and redemption of a WAT ticket.*

1. Issuing – the birth of a WAT ticket

A *drawer* issues a WAT ticket by writing on an empty form the name of the provider (*lender*) of the goods or service, the amount of debt¹, the present date, and the drawer’s signature. The drawer gives the ticket to the lender, and in return obtains some goods or service.

2. Circulation – ordinary exchange

The person to whom the WAT ticket was given can become a *user*, and use it for another trading. To do so, the user writes the name of the recipient, as well as their own, on the reverse side of the ticket. The recipient will become a new user, repeating which the WAT ticket circulates among people.

3. Redemption – the return of the WAT ticket

The WAT ticket is invalidated when it returns, as a result of a trade, to the drawer.

There is a security rule to take care of such situations that the drawer fails to redeem their tickets.

Definition 17 (security rule) *In case the drawer fails to redeem his or her ticket, the lender assumes the responsibility for the debt. If the lender fails, the next user takes over. The responsibility follows the chain of endorsements, up to the last receiver themselves.*

¹Typically in the unit kWh, which represents cost of producing electricity from natural energy sources.

3.2.2 Distinctive Features of the WAT System

The WAT System implements many requirements for the P2P barter currency to a significant extent, namely administration-freedom, interference-freedom, part of locality-freedom, plasticity, strategy and moral-hazard resistance, extensibility and coexistence.

Autonomy

Anyone can spontaneously become a member of the WAT System with a sheet of paper if they follow the rules of the WAT Core.

Compatibility

A WAT ticket is compatible with any other WAT tickets in the world, so that the currency system is operable globally, as long as the drawer can be credited.

Extensibility

An *extended part* can be defined in addition to the WAT Core, for a new currency based on the WAT System. One can state, for example, the region, group and duration in which the tickets are usable, as well as the unit in which the debt is quantified.

Security

Each ticket is backed up by the security rule. The longer the chain of endorsements is, the more firmly backed up the ticket is. Therefore the length of the chain of endorsements represents the extent of trust the ticket has gained.

3.3 Problems in Electronizing the WAT System

The author believes that the WAT System will work better if it is electronized and translated onto the digital communication domain for a number of reasons:

1. It will be able to achieve more locality-freedom if it can utilize the Internet and wireless mobile ad-hoc networks.
2. Fault-tolerance can be improved by replication techniques.
3. More strategy-aware and secure design may be possible with help of digital cryptography.

4. Activity-coordination can be facilitated by networking and digital communication tools.

Upon translating the WAT Core onto the digital communication domain, however, we need to make the following changes to the state machine of a WAT ticket and how the system is operated:

1. Trades need to be asynchronously performed.

Intermediate states, such as waiting for acceptance or approval, needs to be introduced.

2. Double-spending needs to be prohibited

As an effective means, the drawer is to be made responsible for guaranteeing that the circulating ticket is not a fraud. This design is incentive-compatible in such a way that it is the drawer who is victimized by double-spending, which increases the amount of their debt.

This implies that every trade has to be approved by the drawer of the involved ticket.

3. A tradeoff between the cost of identities and autonomy needs to be considered.

If identities are expensive (for example, it requires a certificate issued by a government to obtain an identifier), it would be more secure, but less autonomous. Conversely, if identities are inexpensive (for example, anyone can randomly generate their own identifiers), it would be more autonomous, but less secure.

As an effective means, a PGP public key user ID (an e-mail address under the current operation of PGP) is to be used as the identifier of a participant. This identifier can be autonomously generated by the participant themselves, because it does not have to be a valid e-mail address, being just an identification string, and the key pair to which it is bound can be generated freely; but the authenticity of the identifier needs to be validated by the surrounding parties by constructing a web of trust.

We must not lose good properties shown by the WAT System, such as autonomy, compatibility, extensibility and security.

3.4 *i*-WAT: the Internet WAT System

3.4.1 Overview

i-WAT translates the WAT Core and the security rule onto the Internet. It is also usable without Internet-connectivity with help of (wireless) ad-hoc communication.

In *i*-WAT, messages signed in OpenPGP (*i*-WAT messages) are used to implement transfers of an electronically represented WAT ticket (*i*-WAT ticket). An *i*-WAT ticket contains the identification number, amount of debt and public key user IDs of the drawer, users and recipients. Endorsements are realized by nesting PGP signatures as illustrated in Figure 3.2. (In

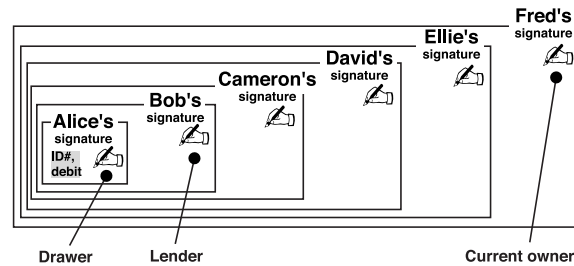


Figure 3.2: Signature chain in an *i*-WAT ticket

the reference implementation, the chain of endorsements is visualized as illustrated in Figure 3.3, using the PGP photo IDs.)

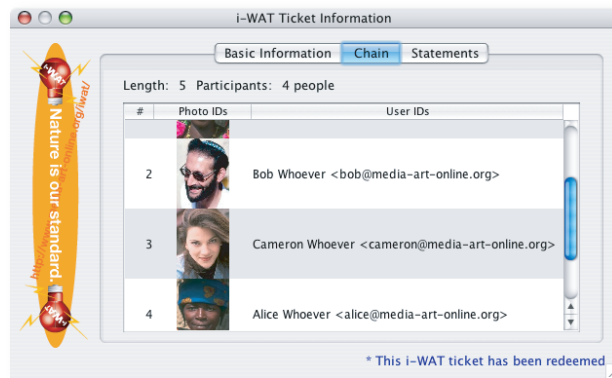


Figure 3.3: Visualized signature chain in the reference implementation

Table 3.1 shows the types of *i*-WAT messages. All *i*-WAT messages are signed by the senders, and are formatted in the canonical form[8] of XML[9]. The messages cause state transfers of a ticket as illustrated in Figure 3.4.

3.4.2 Conditions

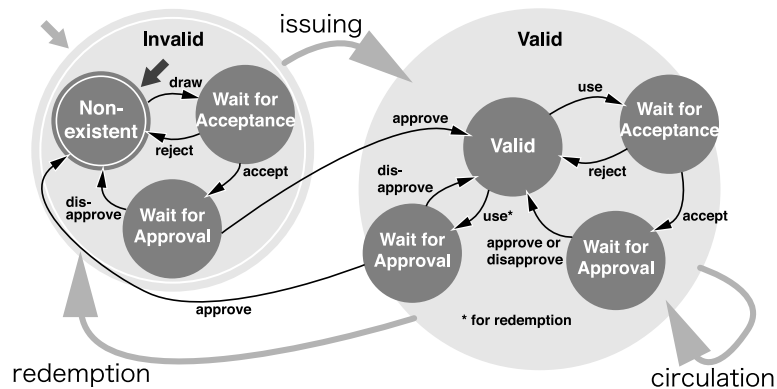
Validation Conditions

The necessity to verify the signatures on *i*-WAT messages implies that the following conditions must have been met when the lifecycle of an *i*-WAT

Table 3.1: *i*-WAT messages

<i>Message</i>	<i>Sender</i>	<i>Receiver</i>	<i>Function</i>
<draw/>	drawer	recipient (lender)	draws an <i>i</i> -WAT ticket.
<use/>	user	recipient	uses an <i>i</i> -WAT ticket.
<accept/>	recipient	drawer and user	confirms readiness to accept the <i>i</i> -WAT ticket once it is validated.
<reject/>	recipient	drawer or user*	rejects an <i>i</i> -WAT ticket.
<approve/>	drawer	user and recipient	validates an <i>i</i> -WAT ticket, and approves the transaction.
<disapprove/>	drawer	user and recipient	denies an <i>i</i> -WAT transaction.

* depending on whether the ticket has just been issued or in circulation, respectively.



* Gray arrows represent WAT state-transfer.

* Black arrows represent *i*-WAT state-transfer.

Figure 3.4: State machine of a WAT/*i*-WAT ticket

ticket is completed.

Definition 18 (validation conditions)

1. *The drawer has the valid public keys of all users appearing in the life-cycle of the ticket.*
2. *Every user has the valid public keys of the drawer and the immediate recipient.*
3. *Every recipient has the valid public keys of the immediate user and the drawer.*

Satisfaction of these conditions are explained in detail in section 4.3.

Reliable Multicast

Because some *i*-WAT messages are sent to multiple entities whose receipts must be agreed, communication channels need to satisfy reliable multicast[33] in a purely conceptual sense (it may be satisfied by a series of unicasts).

Definition 19 (reliable multicast) *Reliable multicast consists of three conditions:*

1. *Validity*
If a correct process multicasts a message m , then some correct process in the group will eventually deliver m , or no process in the group is correct.
2. *Agreement*
If a correct process delivers a message m , then all other correct processes in the group eventually deliver m .
3. *Integrity*
For any message m , every correct process p delivers m at most once, and only if p is in the group and m was previously multicast.

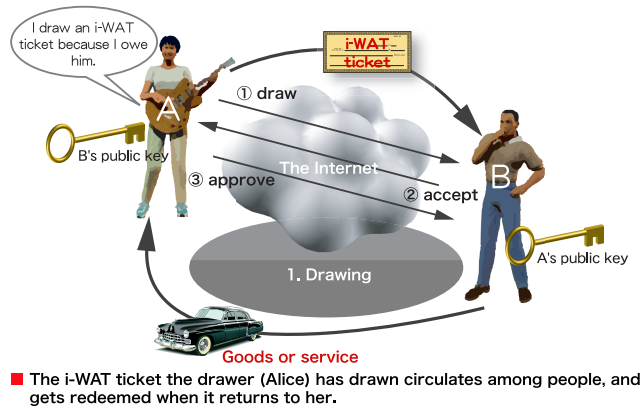
where to *multicast* is to send the same message to all members in the group, and to *deliver* is to put the message forward for processing.

3.4.3 Protocol

(Readers can also find a casual specification of the protocol in section A.2.)

Issuing – the birth of an *i*-WAT ticket (Figure 3.5)

1. The drawer sends a <draw/> message which contains the public key user IDs of the drawer and lender, identification number and amount of debt. This message becomes the original *i*-WAT ticket after the protocol is completed.
2. The lender sends back the content of the message as an <accept/> message.
3. The drawer sends an <approve/> message to the lender.

Figure 3.5: *i*-WAT transaction 1: issuing**Circulation – ordinary exchange (Figure 3.6)**

1. The user adds to the *i*-WAT ticket the public key user ID of the recipient, and sends it to the recipient as a <use/> message. This message becomes a valid *i*-WAT ticket after the protocol is completed.
2. The recipient forwards the content of the message to the drawer and user as an <accept/> message.
3. The drawer verifies the ticket, and sends an <approve/> message to the user and recipient.

Redemption – the return of the *i*-WAT ticket (Figure 3.7)

1. The user sends a <use/> message to the recipient, who equals the drawer.

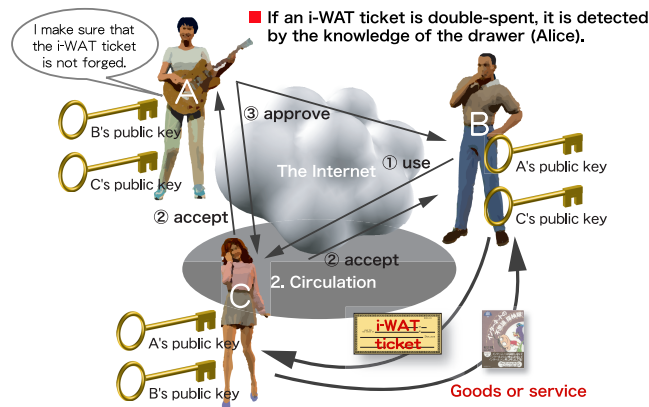


Figure 3.6: *i*-WAT transaction 2: circulation

2. The drawer verifies the ticket, and invalidates it as the debt is now redeemed. The drawer sends an <approve/> message to the user.

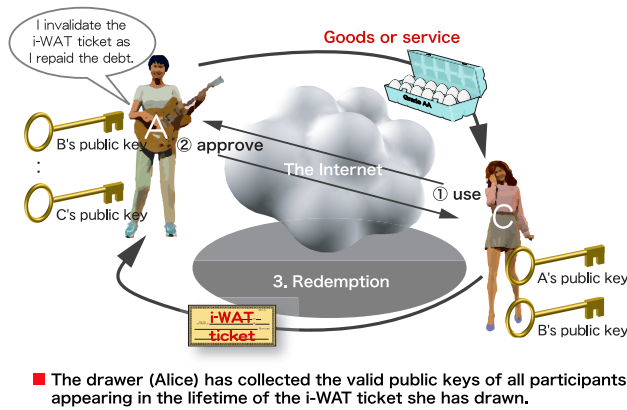


Figure 3.7: *i*-WAT transaction 3: redemption

Chapter 4

Theory

4.1 Overview

The author clarifies how the requirements for the P2P barter currency are to be satisfied in theory.

4.1.1 Approaches to Autonomy

Administration-freedom

This requirement is satisfied primarily by the choice of the basic currency model.

The design of the WAT System allows *on-demand appearance* and *off-demand disappearance* of the WAT tickets to be performed autonomously by the drawers, without necessitating any administrative organizations. *i*-WAT is designed not to break this; its cryptographic authentication is based on webs of trust, instead of a public key infrastructure that would require a single self-certifying authority. Integration allows participants not to depend on key servers to exchange their public keys.

Interference-freedom

This requirement is also satisfied primarily by the choice of the basic currency model.

The WAT System is self-sufficient even in the presence of strategic participants because of its security rule (it does not require external payments to compensate the loss caused by free-riders). *i*-WAT is designed not to break this; the operational cost of *i*-WAT can be negligible in terms of the cost of software, as integration makes it possible to construct an environment for using *i*-WAT solely from free and open source software. An optional use of (wireless) ad-hoc communication implies that possibly the cost of connectivity is not imposed.

Locality-freedom

To satisfy this requirement, the implementation of the WAT Core in *i*-WAT is separated from transportation of its data, so that the layer responsible for conveying data can be switched.

Such a layer needs to use communication channels which satisfy reliable multicast in a purely conceptual sense. Available non-proprietary messaging protocols over the Internet, such as e-mail (SMTP[64]: Simple Mail Transfer Protocol) and instant messaging (XMPP[69, 70]: Extensible Messaging and Presence Protocol), suffice effectively to implement the conceptual reliable multicast, by resending messages if necessary. So do wireless protocols such as CCS[41] (Content Cruising System), which may implement it more efficiently through direct utilization of the shared medium of communication.

This requirement also suggests that rendezvous is possible over physically distant places, which would require appropriate expansions of the webs of trust. This is explained in detail in section 4.3.

4.1.2 Approaches to Safety

Idempotence

Validation of each trade by the drawer should work as a protection for both spurious or repetitive messages and double-spending. In particular, double-spending can be automatically detected and rejected by the drawer.

Fault-tolerance

Two important classes of data in operation of *i*-WAT are key rings (public keys, secret keys and the trust database associated with them) and *i*-WAT tickets.

Public keys are replicated within the webs of trust, which makes restoring them highly possible. In addition, depending on their algorithms, they may be reconstructible from the corresponding secret keys.

One of the worst-case scenarios for a fault in *i*-WAT is that someone's secret key is lost or compromised. In *i*-WAT, the identifier of a participant is their public key user ID instead of the ID of the key itself, which is a hash value of the key. The fault of a lost secret key can be masked by generating a key pair with the same user ID, and reconstructing the web of trust.

Another worst-case scenario is that a drawer loses the data of an *i*-WAT ticket they have issued, which is the official copy of the ticket in the system. In circulation of an *i*-WAT ticket, the data is replicated in three entities: the drawer, the user and the receiver. The loss of data is also maskable by restoring the data from replicated copies from other entities.

Plasticity

This requirement is again satisfied primarily by the choice of the basic currency model.

As mentioned earlier, the WAT System is self-sufficient even in the presence of strategic participants that leave without repaying their debt, because of its security rule. This means that some part of the system can be detached without breaking the correctness of the rest of the system. Moreover, the WAT System can be started spontaneously by a minimal set of autonomous participants: two entities trusting each other. Which makes it virtually impossible to make the system extinct; *i*-WAT is designed not to break these.

There are some obstacles to consider:

1. The security rule may put unpaid debt over someone already heavily in debt, causing a chain reaction of defaults.

This effect is simulated in section 6.1.

2. It is not apparent how the security rule can be enforced, although it can be implemented in the data structure (section 5.2.3).

This is more of a social issue on which more work is needed.

Privacy

All validation relations in the validation conditions of *i*-WAT are reciprocal. Therefore, the senders of *i*-WAT messages necessarily have the valid public keys of the recipients, which makes it possible for the senders to encrypt the messages so that their intended recipients only can read the content of the messages.

There are some obstacles to consider:

1. An *i*-WAT ticket contains the record of all past trades in which it is involved. There is no guarantee that later participants will encrypt their *i*-WAT messages, as there is no clear incentive to do so, and there is a chance that the secrecy of a trade is revealed to some eavesdroppers.

To counteract this obstacle, items for trades are mentioned only in the header part of *i*-WAT messages, not in the data of *i*-WAT tickets themselves. The data contains the promise made by the drawer, which they may rather want to advertise.

2. A traffic analysis is still possible. Encrypting the content of messages alone cannot achieve anonymity if it is required by some participants.

The author argues that this obstacle can be counteracted by the layer responsible for conveying data, which can apply existing techniques such as onion routing[65] to achieve anonymity. This should not be a problem of the exchange core.

Integration prevents unnecessary levels of exposures of e-mail addresses (which are used as public key user IDs) and PGP photo IDs by providing a P2P way of exchanging public keys instead of having to use the public key servers.

Strategy-resistance

This requirement is satisfied through incentive techniques made possible by the semantics of *i*-WAT.

Satisfaction of this requirement is explained in detail in sections 4.4~4.7.

Moral-hazard resistance

This requirement is satisfied by presenting apparent risks to the participants instead of concealing them.

Satisfaction of this requirement is also explained in detail in sections 4.4~4.7.

4.1.3 Approaches to Integration

Openness

This requirement is satisfied by making the specifications open.

The author intends to make the documentations of specifications publicly available. The author has already been distributing the reference implementation under GNU GPL[24] (General Public License) which allows anyone to study and modify the software, and to redistribute modified software under the same license.

This dissertation has an appendix explaining the protocols casually (chapter A).

Extensibility

This requirement is satisfied primarily by the choice of the data description language, which is XML[9], as well as the separation of the data structures from how they are conveyed.

Coexistence

The choice of the basic currency model helps satisfaction of this requirement.

As described earlier, a WAT ticket is compatible with any other WAT tickets in the world, which makes interconnection of two systems based on the WAT System easy. Moreover, the WAT System allows a ticket to be freely associated with any values, including those of exchange mediums of totally different nature; *i*-WAT should be able to utilize these flexibilities.

Satisfaction of this requirement is explained in detail in section 4.11.

Activity-coordination

This requirement is satisfied by building a platform of activity-coordination within whose framework *i*-WAT is operated as one of applications.

The platform provides frameworks to rendezvous, to exchange public keys and to share hypertexts, along with other increasing number of features. The framework to exchange public keys is explained in detail in section 4.9.

4.2 Total Architecture

The author has modeled the total architecture for building an autonomous distributed collaborative systems on top of the exchange mechanism by the P2P barter currency. It has been modeled as five layers of functionalities as illustrated in Figure 4.1. Table 4.1 describes the role of each layer.

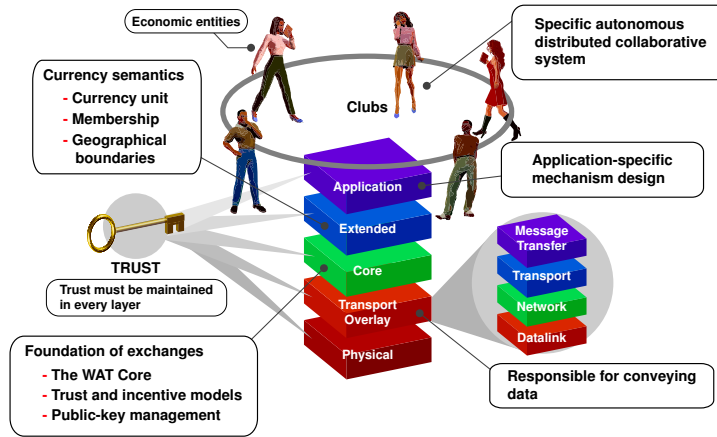


Figure 4.1: Five-layer model

Table 4.1: Roles of layers in the five-layer model

<i>Layer</i>	<i>Role</i>
Physical Layer	Physical connection.
Transport Overlay Layer	Conveyance of data.
Core Layer	The exchange mechanism.
Extended Layer	The semantics of exchanges.
Application Layer	Application-specific mechanism design.

Section 4.12 describes an example of a specific autonomous distributed collaborative system which deals with exchanging information as well as physical resources and labors.

4.3 *i*-WAT and the PGP Trust Model

The author clarifies the trust model of *i*-WAT, and investigates how it is related with that of PGP. To implement the model by dynamically building an appropriate web of trust, the author claims that it would suffice if the behaviors of participants satisfy some specific properties.

4.3.1 *i*-WAT Trust Model

Let us define that $t(x)$ is an *i*-WAT ticket t drawn by x , $\mathcal{U}_{t(x)}$ is the set of users throughout the lifecycle (up to redemption) of $t(x)$, and $y \xrightarrow{t(x)} z$ denotes that y gives $t(x)$ to z as a result or promise of a trade.

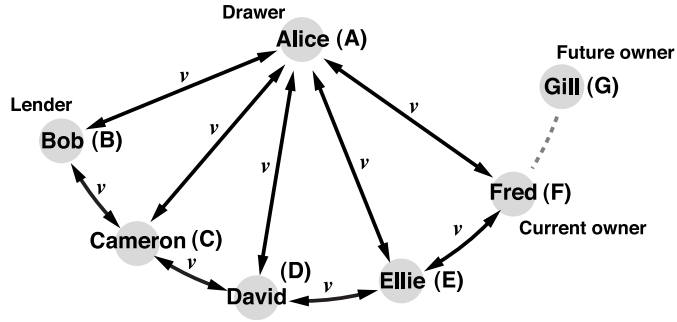


Figure 4.2: *i*-WAT trust model

The *i*-WAT trust model is a definition of *mutually validating relation* $\overset{v}{\leftrightarrow}$ over a network of participants.

Definition 20 (*i*-WAT trust model) for every $t(x)$,

1. for all y such that $y \in \mathcal{U}_{t(x)}$, $x \overset{v}{\leftrightarrow} y$
2. for all y, z such that $\{y, z\} \subseteq \mathcal{U}_{t(x)}$, $y \overset{v}{\leftrightarrow} z$ if $y \xrightarrow{t(x)} z$

Figure 4.2 illustrates the model by an example. This model is naturally induced from the necessity for the participants to validate *i*-WAT messages. This is a formalization of the validation conditions (Definition 18).

4.3.2 Spinning the Web of Trust – Preconditions

If the PGP trust model over the network of participants does not readily support the above model, the model needs to be implemented by dynamically building an appropriate web of trust. In order to do so, we claim that it suffices (but not necessitates) if the behaviors of the participants satisfy the following properties.

Property 5 (mutual signing by knowing)

for every x and y ,

- $x \overset{s}{\leftrightarrow} y$ if x knows¹ y

Intuitively, this states that any two mutual acquaintances sign the public keys of each other.

Property 6 (mutual signing by participation)

for every $t(x)$,

- for all y such that $y \in \mathcal{U}_{t(x)}$, $x \overset{s}{\leftrightarrow} y$

Intuitively, this states that the drawer and a user sign the public keys of each other.

Property 7 (mutual full trust by participation)

for every $t(x)$,

1. for all y such that $y \in \mathcal{U}_{t(x)}$, $x \in \mathcal{T}_y \wedge y \in \mathcal{T}_x$
2. for all y, z such that $\{y, z\} \subseteq \mathcal{U}_{t(x)}$, $y \in \mathcal{T}_z$
if $y \xrightarrow{t(x)} z$

Intuitively, this states that the drawer and a user are confident about each other, and a recipient is confident about the corresponding user, that their correspondents have an excellent understanding of key signing. They need to reflect such views in their PGP trust databases.

Also we assume that GnuPG's default values are used for variables f , m and h .

4.3.3 Spinning the Web of Trust – Case Studies

We justify the above claim by case studies. Throughout the studies, the network of participants in Figure 4.2 is used as an example. It is assumed that no external source of information is available.

Our goal is to show that the *i*-WAT trust model is satisfied in every stage of trades starting from likely initial states, i.e., a joining party knows someone in the network of participants, if the properties explained in section 4.3.2 are satisfied by the participants.

The statement that follows each claim is both a casual proof and a procedure to achieve the goal.

¹In the context of this dissertation, *knows* relation is defined to be symmetrical, i.e., y knows x if x knows y .

Issuing

The goal is to form the initial network of participants between Alice and Bob.

Claim 1 $A \overset{v}{\leftrightarrow} B$ results if and only if Alice knows Bob.

In case Alice knows Bob

1. By *mutual signing by knowing*,

$$A \overset{s}{\leftrightarrow} B$$

2. By the definition 1a of *PGP trust model*,

$$A \overset{v}{\leftrightarrow} B$$

In case Alice does not know Bob There is no definition or property available to deduce $A \overset{v}{\leftrightarrow} B$.

Circulation

The goal is to let Gill join the existing network of participants.

Claim 2 $G \overset{v}{\leftrightarrow} F \wedge G \overset{v}{\leftrightarrow} A$ results if Gill knows either Fred or Alice, or someone (in \mathcal{T}_G) or some people (in \mathcal{T}'_G) in the network of participants.

In case Gill knows both Fred and Alice

1. By *mutual signing by knowing*,

$$G \overset{s}{\leftrightarrow} F \wedge G \overset{s}{\leftrightarrow} A$$

2. By the definition 1a of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} F \wedge G \overset{v}{\leftrightarrow} A$$

In case Gill knows Fred, but not Alice

1. By *mutual signing by knowing* and *mutual signing by participation*,

$$G \overset{s}{\leftrightarrow} F \wedge F \overset{s}{\leftrightarrow} A$$

2. By *expansion of signing-apart relation*,

$$G \overset{s}{\leftrightarrow} F \wedge G \overset{s}{\leftrightarrow} F \overset{s}{\leftrightarrow} A$$

3. By the definition 1a of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} F \wedge G \overset{s}{\leftrightarrow} F \overset{s}{\leftrightarrow} A$$

4. $F \in \mathcal{T}_A$ and $F \in \mathcal{T}_G$ by the properties 1 and 2 of *mutual full trust by participation*, respectively. Also, the path length between Gill and Alice is shorter than h . Therefore, by the definition 1b of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} F \wedge G \overset{v}{\leftrightarrow} A$$

In case Gill knows Alice, but not Fred

1. By *mutual signing by knowing* and *mutual signing by participation*,

$$G \overset{s}{\leftrightarrow} A \wedge A \overset{s}{\leftrightarrow} F$$

2. By *expansion of signing-apart relation*,

$$G \overset{s}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

3. By the definition 1a of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

4. $A \in \mathcal{T}_G$ and $A \in \mathcal{T}_F$ by the property 1 of *mutual full trust by participation*. Also, the path length between Gill and Fred is shorter than h . Therefore, by the definition 1b of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{v}{\leftrightarrow} F$$

In case Gill knows neither Alice nor Fred The goal can still be met if

1. there is one user x such that $x \in \mathcal{U}_{t(A)}$ who knows Gill and appeared earlier than Fred, and $x \in \mathcal{T}_G$, or
2. there are three users x, y, z such that $\{x, y, z\} \subset \mathcal{U}_{t(A)}$ who all know Gill and appeared earlier than Fred, and $\{x, y, z\} \subseteq \mathcal{T}'_G$.

The proofs for the above two cases are similar; they both involve first establishing $G \overset{v}{\leftrightarrow} A$ by way of someone or some people in the middle, only that the latter is more complex.

Suppose Gill knows Cameron, David, Ellie, and marginally trust them.

1. By *mutual signing by knowing* and *mutual signing by participation*,

$$G \overset{s}{\leftrightarrow} C \wedge C \overset{s}{\leftrightarrow} A \wedge A \overset{s}{\leftrightarrow} F$$

2. By *expansion of signing-apart relation*, and by the definition 1a of *PGP trust model*,

$$\begin{aligned} G \overset{v}{\leftrightarrow} C \wedge C \overset{s}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \\ \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F \end{aligned}$$

3. The above also holds if we replace C with D or E . It is given that $\{C, D, E\} \subseteq \mathcal{T}'_G$. Also, the path length between Gill and Alice is shorter than h . Therefore, by the definition 1c of *PGP trust model*,

$$\begin{aligned} G \overset{v}{\leftrightarrow} A \wedge C \overset{s}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \\ \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F \end{aligned}$$

4. $C \overset{v}{\leftrightarrow} A$ by the definition 1a of *PGP trust model*. $C \in \mathcal{T}_A$ by the property 1 of *mutual full trust by participation*. Therefore, by the definition 1b of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

5. Now that Gill and Alice mutually validates their public keys, they can establish $G \overset{s}{\leftrightarrow} A$ by *mutual signing by participation*.

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

6. $A \in \mathcal{T}_G$ and $A \in \mathcal{T}_F$ by the property 1 of *mutual full trust by participation*. Also, the path length between Gill and Fred is shorter than h . Therefore, by the definition 1b of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{v}{\leftrightarrow} F$$

Redemption

The goal is to complete the lifecycle of the ticket in concern without further expanding the existing network of participants.

Claim 3 $G \overset{v}{\leftrightarrow} A$ results without building the web of trust any further.

1. By *mutual signing by participation*,

$$G \overset{s}{\leftrightarrow} A$$

2. By the definition 1a of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A$$

4.3.4 Justification of the Preconditions

We casually explain how the preconditional properties are supported by the natural behaviors of participants.

Mutual Signing by knowing

If two parties know each other (well enough), it should be possible to safely exchange the fingerprints² of their public keys. Therefore this is only a question of the communication cost.

Mutual Signing by Participation

Because it becomes easier for other participants to join the circle of friends around an *i*-WAT ticket if this property is met, both the drawer and user have incentives to sign each other's public keys after properly validating them.

Mutual Full Trust by Participation

The participants are motivated to fully trust their correspondents in the context of public key signing by the same incentives as the above. Also, they are disincentivized to be negligent of the precautions for signing public keys, in order to protect themselves from possible attacks by impostors.

4.4 General Incentive Model

The author models a series of trades with an *i*-WAT ticket as a sequential game with incomplete information.

4.4.1 Generalized Ticket Value

The author makes a generalization to the value of a WAT/*i*-WAT ticket.

Definition 21 (Generalized value) *The value of a WAT/*i*-WAT ticket is expressed as a tetrad $\langle V_0, V_m, V_x, f \rangle$ presented by the drawer, where V_0 is the face value (initial value) of the ticket, V_m is the minimum value, V_x is the maximum value, and $f(t)$ is the differentiation (derivative) of a function of time $F(t)$. V_m/V_x are set to be \perp/\top respectively if those values are not applicable.*

The effective value V_t of a ticket at time t is given by the following equation:

$$V_t = \min(\max(\int_0^t f(t)dt + V_0, V_m), V_x)$$

²A fingerprint is a hash value of a key so that the key's identity can be checked with small cost.

This is a generalization to allow the value of a ticket to vary over time, limited by some minimum/maximum values. Typically, it holds that either $f(t) = 0$ for all t , $f(t) < 0$ for all t or $f(t) > 0$ for all t .

Definition 22 (regular ticket) *A WAT/ i -WAT ticket is a regular ticket if and only if $f(t) = 0$ for all t .*

Definition 23 (reduction ticket) *A WAT/ i -WAT ticket is a reduction ticket if and only if $f(t) < 0$ for all t .*

The incentive mechanism for reduction tickets have been discussed in [80, 81].

Definition 24 (multiplication ticket) *A WAT/ i -WAT ticket is a multiplication ticket if and only if $f(t) > 0$ for all t .*

The incentive mechanism for multiplication tickets have been discussed in [79].

4.4.2 Notations and Preconditions

Participants

Users are denoted as W (for WAT friends) indexed by the order of their appearance: drawer = W_0 , lender = W_1 , ..., current recipient = W_n . For the sake of argument, there assumed to be $n+1$ unique participants, and the webs of trust around them are built from scratch as transactions proceed.

Probability of Defaults

Probability p_i divides W_i into two types: *successful* (appears by probability $1 - p_i$) or *failing* (appears by probability p_i) to redeem the ticket in concern. Note that in WAT/ i -WAT, the responsibility follows the chain of endorsements if the drawer fails to redeem, by the security rule.

Assuming that all participants behave rationally, they *will* default if they are impostors of real or imaginary persons because they can get away with unpaying the cost of trust. Therefore the probability that W_i 's identity is forged is regarded at most p_i .

Timing of Usage

The time at which W_i uses the ticket is regarded i to simplify reasoning. This means that the time is not evenly distributed in the model. Still, for any *reduction* tickets, it holds that $V_i < V_{i-1}$, and for any *multiplication* tickets, it holds that $V_i > V_{i-1}$, where $i > 0$.

Redemption takes place at time r .

Utility of Exchange

There assumed to be some utility of having an exchange medium instead of having specific goods or unutilized services. This utility for W_i is denoted as UX_i .

UX_0 is a special case, where the value is divided into utility of spending UX_0^S and utility of earning (redemption) UX_0^E , to reflect the fact that these events are not adjacent in the time line.

Cost

Cost of trust Cost to rebuild trust relationships for W_i is CT_i . The cost includes that of *whitewashing*, or that one disappears and assumes a new identity. It is assumed that this cost does not vary in a large extent among participants, and is generally worth more than a value of a ticket. These assumptions should be justified by the fact that the *i*-WAT trust model requires construction of a *web of trust*[76], which requires that a new participant must know someone in person in the circle of friends around the *i*-WAT ticket.

Cost of lazy approval Cost of lazy approval by W_0 for a recipient W_i is denoted as CL_i . It is apparent that this cost exists for a *reduction* ticket, whose value is reduced over time. The cost exists for other types of tickets too, because it affects the usability of the ticket in concern; the ticket will not be usable by W_i until W_0 approves the transaction in which W_i received the ticket.

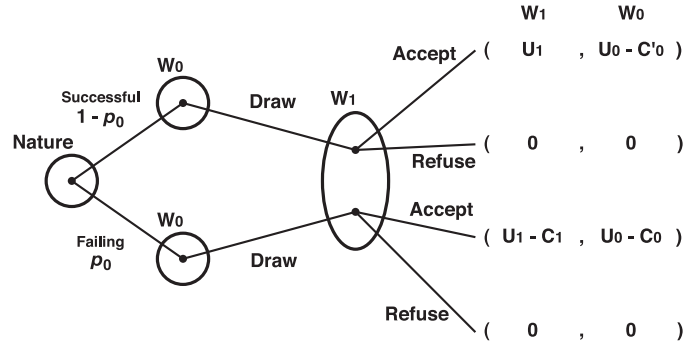
Laziness of W_0 is assumed to be observable from others. This assumption is justifiable by a software design; participants can observe how often W_0 becomes online in an *i*-WAT-enabled presence-sharing system.

Cost of premature redemption Cost of unexpectedly early redemption for W_0 is denoted as CP_0 . Note that W_0 is incentivized to delay redemption even for *multiplication* tickets, which will often be used to control the timing of redemption by giving users incentives to wait.

Cost of communication Communication cost is negligible for *i*-WAT, which is the reason why the WAT System was electronized and made usable on the Internet and (wireless) ad-hoc networks.

Accounting

The sum of effective values of all tickets issued by W_0 in circulation is denoted as $\sum V$. This information is assumed to be made available to all prospective participants. Feasibility of this is discussed in section 4.8.



$$U_1 : UX_1 + V_1 - V_0 - CL_1, \quad C_1 : V_r(1 - p_1) + CT_1p_1$$

$$U_0 : UX_0^S + V_0, \quad C_0 : \frac{V_r CT_0}{\sum V}, \quad C'_0 : V_r + CP_0 - UX_0^E$$

* $V_r = V_1$ and $p_1 = 0$ if W_1 is the last user

Figure 4.3: Game tree for issuing a ticket

Since the cost of trust CT_0 is to be applied just once when W_0 white-washes their identities, W_0 can minimize the effectiveness of the cost by issuing as many tickets as they can and then go on to default (see section 4.7.6). Therefore prospective lenders are interested in this information.

4.4.3 Game Trees

It is often useful to draw *game trees* when analyzing transactions as games. A *game tree* is a graph consisting of players' decision points as nodes, which are connected in the order of their occurrences. Each player has an *information set*, or a set of decision points from which they can choose an action. In the end of the graph, the gains of all players are drawn as leaves.

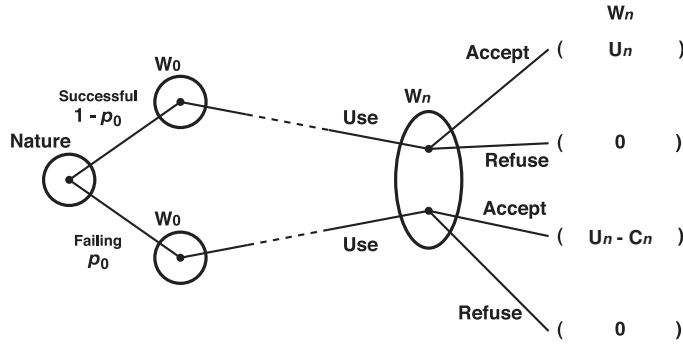
In the figures to follow, types of participants are not made explicit in the trees except for those of W_0 , which are distinguished by probability p_0 .

Payoffs for issuing

Figure 4.9 shows a game tree for issuing an *i*-WAT ticket.

The first player is the nature who chooses between two types of W_0 as the drawer: *successful* or *failing* to redeem the ticket. These types appear by probabilities of $(1 - p_0)$ and p_0 , respectively, for reasons either situational or strategic which are not distinguishable by other participants.

The lender W_1 has an information set in which the player is uncertain about W_0 's type. Depending on the player's belief, W_1 chooses to either accept or refuse the ticket presented by W_0 .



$$U_n : UX_n + V_n - V_{n-1} - CL_n$$

$$C_n : (V_r(1 - p_n) + CT_n p_n) \prod_{i=1}^{n-1} p_i$$

* $V_r = V_n$ and $p_n = 0$ if W_n is the last user

Figure 4.4: Game tree for circulating a ticket

Inside parentheses are the gains of W_1 and W_0 in each combination of W_0 's type and W_1 's action.

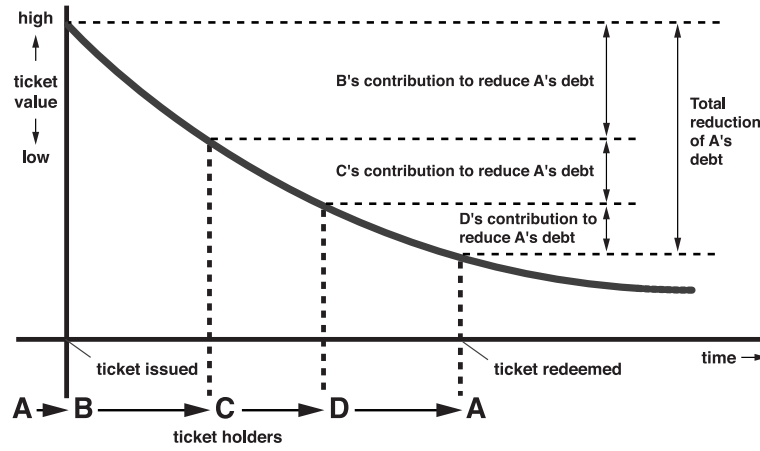
1. If W_1 chooses to accept the ticket
 - W_1 's expectation is $U_1 - C_1 p_0$
 - W_0 's expectation is $U_0 - C'_0(1 - p_0) - C_0 p_0$
2. If W_1 chooses to refuse the ticket
 - Both W_0 and W_1 gain or lose nothing.

The utility UX_1 depends in large part on whether the ticket will be accepted by W_2 or not. It is also an important factor for minimizing $|V_1 - V_0|$ for a *reduction* ticket, in which case both W_0 and W_1 wish V_r to be zero. In case of a *multiplication* ticket, W_1 will typically wait until the effective value reaches V_x , and then use the ticket against W_0 for both maximizing their gain $V_1 - V_0$ (in case of successful W_0) and minimizing their loss to V_0 (in case of failing W_0).

In any case, p_0 is an important factor for W_1 to make a decision.

Payoffs for circulation

Figure 4.10 shows a game tree for circulating an *i*-WAT ticket. The tree is an extension to Figure 4.9.

Figure 4.5: Meaning of a *reduction* ticket

1. If W_n chooses to accept the ticket
 - W_n 's expectation is $U_n - C_n p_0$
2. If W_n chooses to refuse the ticket
 - W_n gains or loses nothing.

If n is small, W_n is interested in the trustworthiness of all participants W_i where $0 \leq i < n$. Since $\prod_{i=1}^{n-1} p_i$ approaches zero as n increases, W_n will be indifferent of the type of W_0 if n is sufficiently large; they will tend to accept the ticket.

This may lead to a moral hazard, but still W_n will be interested in maintaining the trust model of i -WAT as described in the following sections.

4.5 ROT: Reduction Over Time

4.5.1 Concept

Reduction of the value of a ticket means that the drawer's debt is reduced. The cost of reduction is first admitted by the lender who credits the drawer, and then shared among the endorsers as illustrated in Figure 4.5. The amount of the total reduction is manifested to the drawer upon redemption. By deferring redemption, participants can help easing the burden of the drawer. At the same time, they are helped by the utility of the ticket the drawer has issued (*mutual help among peers*).

4.5.2 Incentive-Compatibility of the Design

We show that the design of *reduction* tickets is incentive-compatible by modeling the series of transactions with such a ticket as a sequential game with incomplete information.

Predictions

Our analysis resulted in a prediction that the following properties will hold.

Property 8 (rapid circulation) *Unless $V_t = V_m$, participants are incentivized to minimize the duration of holding a reduction ticket, and they are disincentivized to use it against the drawer.*

Property 9 (vanishment equilibrium) *If $V_m = 0$, a special case of perfect Bayesian Nash equilibrium is achieved; under the belief that no one defaults (because rapid circulation suggests that the value at redemption will be zero), all participants have no incentive to refuse the reduction ticket.*

Incentives and rationality Existence of the following set of incentives is assumed as the common knowledge among participants.

1. The drawer is interested in maximizing their own benefit.
2. Other participants are interested in minimizing their losses and at the same time participating in maximizing the drawer's benefit. In other words, they are willing to help, but unsure about the extent to which they want to sacrifice their own benefits.

It is assumed that all participants are rational.

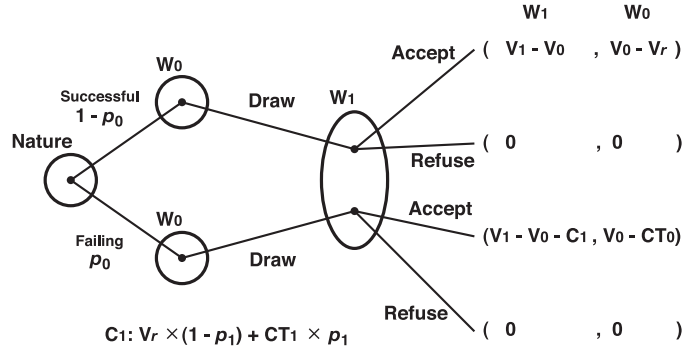
Game Trees

Payoffs for issuing Figure 4.9 shows a game tree for issuing a *reduction* ticket.

The first player is the nature who chooses between two types of W_0 as the drawer: *successful* or *failing* to redeem the ticket. Again, these types appear by probabilities of $(1 - p_0)$ and p_0 , respectively, for reasons either situational or strategic which are not distinguishable by other participants.

The lender W_1 has an information set in which the player is uncertain about W_0 's type. Depending on the player's belief, W_1 chooses to either accept or refuse the ticket presented by W_0 .

Inside parentheses are the gains of W_1 and W_0 in each combination of W_0 's type and W_1 's action.



* $V_r = V_1$ and $p_1 = 0$ if W_1 is the last user

Figure 4.6: Game tree for issuing a *reduction* ticket

1. If W_1 chooses to accept the ticket
 If W_0 's type is successful, W_1 gains $V_1 - V_0$ which is negative, and W_0 gains $V_0 - V_r$ which is positive. Otherwise W_1 gains $V_1 - V_0 - C_1$, which is still negative, where C_1 is the cost of default for W_1 expressed as follows:

$$C_1 : V_r \times (1 - p_1) + CT_1 \times p_1$$

which intuitively states that the player has either to pay equivalent of V_r or to lose their trust, depending on the probability of his or her failure p_1 . W_0 gains $V_0 - CT_0$ if the player's type is failing, which is assumed to be negative.

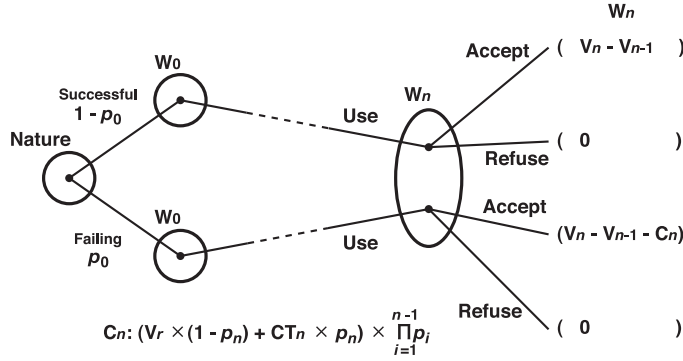
2. If W_1 chooses to refuse the ticket
 Both W_0 and W_1 gain or lose nothing.

Payoffs for circulation Figure 4.10 shows a game tree for circulating a *reduction* ticket, which is an extension to Figure 4.9.

1. If W_n chooses to accept the ticket
 If W_0 's type is successful, W_n gains $V_n - V_{n-1}$ which is negative. Otherwise the player gains $V_n - V_{n-1} - C_n$, which is also negative, where C_n is the cost of default for W_n expressed as follows:

$$C_n : (V_r \times (1 - p_n) + CT_n \times p_n) \times \prod_{i=1}^{n-1} p_i$$

2. If W_n chooses to refuse the ticket
 W_n gains or loses nothing.



* $V_r = V_n$ and $p_n = 0$ if W_n is the last user

Figure 4.7: Game tree for circulating a *reduction* ticket

Analysis

First, we would like to know whether it is rational or not for the lender W_1 to accept a *reduction* ticket. Since no one would want to receive an exchange medium which cannot be used for an exchange, we begin by investigating if W_n would accept the ticket in circulation.

Rational behaviors of W_n The expectation of the gain if W_n chooses to accept the ticket is as follows:

$$\begin{aligned} & (V_n - V_{n-1}) \times (1 - p_0) + (V_n - V_{n-1} - C_n) \times p_0 \\ & = V_n - V_{n-1} - C_n p_0 \end{aligned}$$

It is inevitable that this value is negative, but W_n wants to take part in maximizing W_0 's benefit. Thus W_n seeks a possibility of making the value very close to zero so that the loss is negligible (and the implicit utility of the currency exceeds the loss).

$\prod_{i=1}^{n-1} p_i$ approaches zero as n increases, which makes the cost C_n negligible if n is sufficiently large. W_n can control the value V_n and minimize $V_{n-1} - V_n$ by spending the ticket as soon as possible. To do so, there must be someone willing to accept the ticket.

There are two candidates against whom W_n can use the ticket: the drawer W_0 and the prospective participant W_{n+1} . Since W_n is incentivized to maximize the benefit of W_0 , the player decides to refrain from using the ticket against W_0 unless $V_n = V_r = V_m$, at which time W_0 's gain $V_0 - V_r$ is maximized.

By applying the same reasoning as their own, W_n can infer that W_{n+1} will try to minimize their loss, only that C_{n+1} is even smaller than C_n .

Apparently, W_{n+1} is in a better position than W_n with respect to the cost of default, so that W_n infers that W_{n+1} will accept the ticket if W_n chooses to accept it.

Therefore, there is no strategic reason for W_n not to accept the ticket if n is sufficiently large.

Rational behaviors of W_1 The expectation of the gain if W_1 chooses to accept the ticket is as follows:

$$V_1 - V_0 - C_1 p_0$$

which is also inevitably negative, but W_1 wants to take part in maximizing W_0 's benefit as others do.

W_1 can control the value V_1 and minimize $V_0 - V_1$ by spending the ticket as soon as possible. As W_n reasoned, W_1 can infer that it is likely that $C_1 > C_2$, or W_2 is in a better position than W_1 with respect to the cost of default (unless W_1 's type is failing and CT_1 is small). The question is how W_1 can minimize C_1 .

It is assumed that $V_r < CT_1$, so that W_1 would like both V_r and p_1 to be as small as possible. Since the incentives support that V_r is ultimately V_m , ease of acceptance for W_1 increases as presented value of V_m becomes small, whose lowest possible value is zero.

Therefore W_1 is most likely to choose to accept the ticket if $V_m = 0$.

Rational behaviors of W_0 We would like to know if W_0 would agree to issue a *reduction* ticket whose value is to be reduced down to zero. Intuitively, the answer is yes, as this would maximize W_0 's gain $V_0 - V_r$. In fact, it is the only equilibrium we can reach if the ticket is accepted.

Note that CT_0 is applied just once after an incident of default. If we consider existence of other tickets issued by W_0 , the cost of default for W_0 should be more precisely expressed as $V_0 - \frac{CT_0}{N}$, where N is the number of W_0 's tickets in circulation. It is possible that $V_r > \frac{CT_0}{N}$ if N is sufficiently large. To eliminate the risk of default, V_r needs to be zero.

Therefore, desires of W_0 and W_1 match, and $V_m = 0$ results in an equilibrium where everyone believes that no one defaults (it is impossible to do so), and no one has incentive to refuse the ticket.

Hazards

We consider two probable hazards caused by the design: moral and timing hazards (the problems of colluding will be discussed later).

Moral hazard The prediction that the most stable value for V_m is zero implies no risk for W_0 to issue new *reduction* tickets.

To prevent W_0 from excessively issuing tickets, the group of lenders who take the role of W_1 must share information among one another about the sum of effective values of W_0 's tickets in circulation. Since it must be a common knowledge that W_0 is in need to begin with, this should not be difficult to achieve.

Timing hazard There is a risk that circulation may be stalled by negligence of W_0 in their role for approving transactions. However, we can show that it is to W_0 's own benefit to maintain the liveness of their tickets.

If W_0 is late to respond to the request for approval, the prospective transaction is delayed. It is the recipient W_{n+1} 's interest that transaction is performed as quickly as possible, otherwise their loss $V_n - V_{n+1}$ cannot be minimized. Meanwhile, W_0 is not affected by their own laziness because the effective value will not decrease further after redemption as the ticket itself will disappear. When likelihood of acceptance is in question, if W_n needs to choose from the two, their natural choice is to ask W_0 for redemption, which is against W_0 's interest.

Therefore, being lazy is to risk early redemptions, and W_0 is incentivized to respond quickly.

4.6 MOT: Multiplication Over Time

4.6.1 Concept

Multiplication of the value means that the drawer's debt is increased. The increased value is first potentially received as a premium by the lender who credits the drawer, and then shared among the endorsers as illustrated in Figure 4.8. The amount of the total increase is manifested to the drawer upon redemption. By deferring redemption, the participants can maximize their gains. Intuitively, this would motivate the participants to receive the ticket (and to provide something in return).

4.6.2 Incentive-Compatibility of the Design

Our analysis resulted in the following predictions.

Predictions

Property 10 (deferred redemption) *If the lender W_1 accepts the ticket, they are likely to use it against the drawer W_0 themselves, and to defer it until the effective value reaches V_x .*

Property 11 (no strategic default) *The drawer W_0 is incentivized to successfully redeem the multiplication tickets they issue. The upper bound of V_x is CT_0 .*

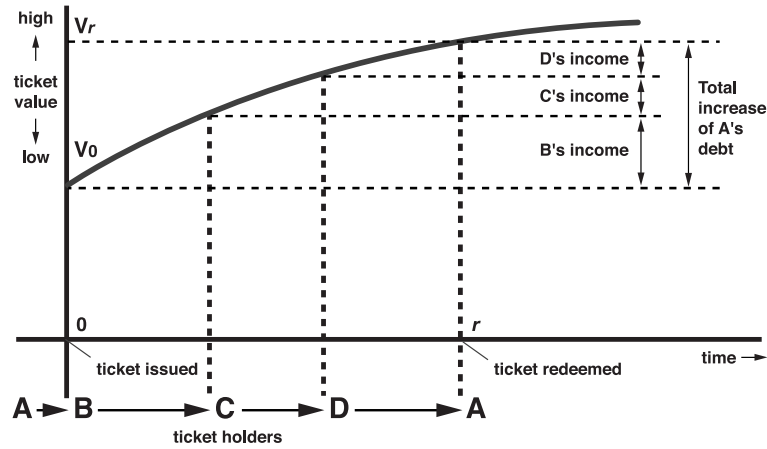


Figure 4.8: Meaning of a *multiplication* ticket

Property 12 (acceptance criterion) *The lender W_1 is likely to accept the ticket if $1 - \frac{V_0}{V_x} > p_0$. The drawer W_0 will try to increase the chance by keeping p_0 low.*

Property 13 (ease of flow) *If the lender W_1 is willing to take the risk, later participants W_n are likely to accept the ticket where n is sufficiently large.*

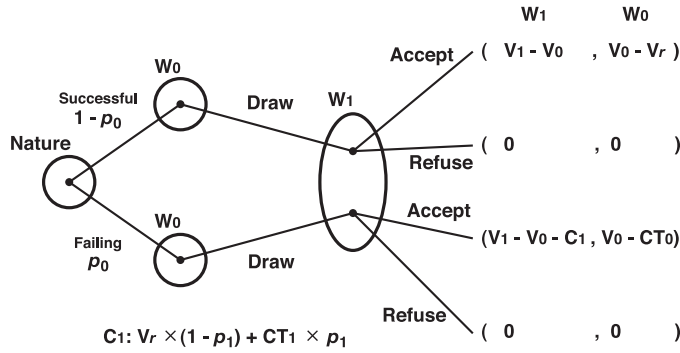
Game Trees and Analysis

Figure 4.9 shows a game tree for issuing a *multiplication* ticket, which is the same as that for a *reduction* ticket.

Deferred Redemption If W_0 's type is *successful*, W_1 can maximize the gain by choosing the largest possible value for V_1 , which is V_x . If W_0 's type is *failing*, W_1 can minimize the loss to be V_0 by not forwarding the ticket to the third person. Therefore, to maximize the gain and to minimize the loss, W_1 chooses to wait until the effective value reaches V_x and tries to use it against W_0 . In the discussions to follow, it is assumed that $V_1 = V_r = V_x$.

No Strategic Default If $V_r \leq CT_0$, then there is no reason for W_0 to default. If $V_r > CT_0$ (thus $V_x > CT_0$), then W_1 knows that W_0 is likely to default. To prevent the loss of V_0 , W_1 would not accept the ticket if $V_x > CT_0$. Therefore the upper bound for V_x of an acceptable ticket is CT_0 .

A strategic default is still possible if there are more than one *multiplication* tickets W_0 have issued in circulation, and W_0 disappears and assumes a new identity (*accumulation attack*): CT_0 may be smaller than the sum of V_x 's. Prevention of this is discussed in section 4.8.



* $V_r = V_1$ and $p_1 = 0$ if W_1 is the last user

Figure 4.9: Game tree for issuing a *multiplication* ticket

Acceptance Criterion If W_1 chooses to accept a ticket, W_1 's expectation is:

$$V_1 - V_0 - C_1 p_0$$

where V_1 and C_1 are both ultimately V_x given that $p_1 = 0$ (W_1 is the last user). W_1 chooses to accept the ticket if this value is greater than that of choosing to refuse it, which is zero. Therefore, the criterion is expressed in the following expression:

$$1 - \frac{V_0}{V_x} > p_0$$

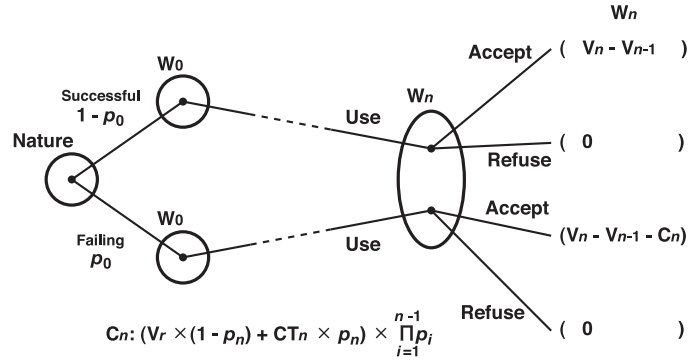
There are three variables: V_0 , V_x and p_0 , which W_0 can manipulate to increase the chance of acceptance. V_0 is decided by W_0 's need, and V_x is bounded by C_{T_0} . Therefore in actuality W_0 can only decrease p_0 .

Ease of Flow Figure 4.10 shows a game tree for circulating a *multiplication* ticket, which is the same as that for a *reduction* ticket. Since $0 \leq p_i \leq 1$, $\prod_{i=1}^{n-1} p_i$ approaches zero as n increases, which makes the cost C_n negligible for W_n . Therefore W_n can choose to accept the ticket regardless of W_0 's type if n is sufficiently large.

4.7 Strategies and Moral Hazards

4.7.1 Overview

This section describes the protection against strategies and counteractions against moral hazards which have been casually discussed in [78]. Table 4.2 shows the list of misbehaviors in concern.



* $V_r = V_n$ and $p_n = 0$ if W_n is the last one

Figure 4.10: Game tree for circulating a *multiplication* ticket

Table 4.2: Possible misbehaviors and the imposed risks to the subjects

Name	Description	Risk to the Subject
Compromised secret	The subject's secret key is compromised or lost.	Cost of trust/ Entrapment
Evidenceless signing	Signs public keys without checking their validity.	Impostors/ Suspect for collusion
Evidenceless full trust	Gives full trust to someone without knowing them.	Impostors/ Suspect for collusion
Excessive issuing	Issues an excessive amount of tickets.	Defaults → cost of trust/ Premature redemptions
Lazy approval	Be late in approving transactions.	Premature redemptions
Defaults	Defaults upon redemption.	Cost of trust
Empty promise	Receives the ticket and escapes without providing promised goods or service	Cost of trust

A case of someone receiving goods or service and escaping without providing a ticket is not discussed because it does not involve a successful i -WAT transaction, and there can be no proof of the incident within the context of the WAT Core (operational solutions need to be pursued).

Double-spending is also excluded from the list because its detection can be automated (it is in our reference implementation), and W_0 has no incentive to turn off such a software feature.

4.7.2 Safety and Risks

The most important safety of WAT/ i -WAT is that the debt does not disappear without redemption.

There are apparent risks for a participant that the drawers of the acquired tickets go bankrupt, and by the security rule of the system, the debts are transferred up to the participant in question. This risk for a ticket is expressed as an expected cost C_n for the n th receiver of the ticket, where the probability of the drawer going bankrupt is p_0 , that for the i th receiver is p_i , the cost of regaining trust after going bankrupt is CT_i , and the value of the ticket in concern at the time of redemption is V_r :

$$C_n = (V_r \times (1 - p_n) + CT_n \times p_n) \times \prod_{i=0}^{n-1} p_i$$

Expected gain G_n for the n th receiver is expressed as follows, where the effective value of the ticket when the i th receiver uses it is V_i :

$$G_n = V_n - V_{n-1} - C_n$$

Which is necessarily negative unless it is a *multiplication* ticket. To minimize these cost, some evasive actions (*EVs*) and a graceful action (*GRs*) are possible as shown in Table 4.3.

Table 4.3: Evasive and graceful actions

<i>Name</i>	<i>Action</i>
EV1 (elimination)	Always try to use a ticket the partner has drawn if there is one.
EV2 (stretch)	Always try to receive a ticket whose chain of endorsement is longer than those of others.
EV3 (matchmaking)	Prefer selecting a partner among the drawers of acquired tickets.
EV4 (forwarding)	Always try to use a ticket whose loss of value will be greater than those of others if not used now.
GR1 (deferring)	Always try to avoid using a ticket against its drawer if the variance of its value over time has not stopped.

EV1 (elimination) makes sure that a ticket is eliminated whenever there is a chance. EV2 (stretch) makes sure that n is reasonably large. EV3 (matchmaking) increases the chance of eliminating a ticket. EV4 (forwarding) for the case of *reduction* ticket is to minimize the loss by variance of the ticket values over time. EV4 for the case of *multiplication* ticket is to maximize the gain by variance of the ticket values over time, because the loss will be negative and it will be unlikely that the ticket is used until variance stops and no more gain can be expected. GR1 (deferring) for the case of *reduction* ticket is to help the drawer reducing their debt. GR1 for the case of *multiplication* ticket is to minimize the loss caused by bankruptcy of the drawer; if one has used the ticket in spite of GR1 and the drawer defaulted, they will have to owe more than the values they received and used. GR1 is graceful for the drawers even in the case of multiplication tickets because such tickets are often used for setting up a timer for a deferred redemption as described in [79].

Effects of EV1~EV3 are simulated in section 6.1.4. Effects of EV4 and GR1 are simulated in section 6.1.5.

4.7.3 Sloppy Key Management

i-WAT uses public key cryptography as a protection against impostors. Therefore, failing to follow the good practice is considered a moral hazard, as speeding at intersections with traffic signals or stop signs would be. Keeping the good practice, on the other hand, maintains the trust model, and prevent offenders from getting away with unpaying the cost of trust.

This section describes how failing to follow the good practice in key management is against the subject's own interest. Discussions at later sections assume that the trust model is maintained.

Compromised Secret

If a secret key is compromised or lost, the key needs to be declared invalid, and replaced with a new one. Since an *i*-WAT ticket records the public key user IDs instead of the identifiers of the keys themselves, replacing the key does not affect the correctness of the data. However, this replacement costs equivalent to CT_i for W_i with the secret key in question because it involves reconstruction of the web of trust. Besides, the compromised key may be used for an entrapment (section 4.7.8).

Evidenceless Signing/Full Trust

If participants sign public keys of others without personally validating them, or if they fully trust other participants without knowing their trustworthiness, there is a risk of allowing impostors of real or imaginary persons in the circle of friends around the *i*-WAT ticket.

Such impostors may perform misbehaviors like an empty promise, by which the signer/truster may be victimized. Or worse, they may be suspected as collaborators of such misbehaviors.

4.7.4 Excessive Issuing

Excessive issuing can mean more debt than W_0 can handle, so that there is a risk of defaults (increased p_0), which discourages both W_0 and W_1 to give birth to a ticket.

Furthermore, since excessive issuing is assumed to be observable from current ticket owners, they would want W_0 to redeem the tickets quickly, in order to avoid W_0 's defaults with the tickets they have. This should be especially true for those tickets whose chains of endorsements are still short. Which means that excessive and intensive issuing attracts premature redemptions.

4.7.5 Lazy Approval

There is a risk that circulation may be stalled by negligence of W_0 in their role of approving transactions.

Let us stand upon W_{n-1} 's view point. If W_0 is late to respond to the request for approval, the prospective transaction is delayed, costing CL_n to W_n which W_{n-1} knows that W_n can predict. Meanwhile, W_0 is not affected by their own laziness because acceptance and approval happen at the same time. When likelihood of acceptance is in question, W_{n-1} 's natural choice is to ask W_0 for redemption.

Therefore, being lazy is to risk premature redemptions, and W_0 is incentivized to respond quickly.

4.7.6 Defaults

W_0 would want to minimize C_0 upon defaults. If V_r can be reduced (as in the case of a *reduction* ticket), there may be no reason to default to begin with. Therefore, the only option for W_0 is to increase $\sum V$ to minimize the effect of CT_0 . However, the value is monitored by all prospective lenders, so that W_0 cannot increase it over a reasonable amount.

4.7.7 Empty Promise

If there is a proof of an empty promise, W_0 can disapprove further transactions with the ticket. If the ticket has not been used further, W_{n-1} can safely become the valid owner of the ticket by a roll back.

The proof of the incident becomes a source of bad reputation for W_n , which can only be whitewashed by paying the cost of trust.

4.7.8 Collusions

Colluded Defaults

There may be a colluded defaults by every W_i where $0 \leq i < n$, so that W_n is victimized. However, the trust model implies that W_n must have needed to know someone in person in the chain of endorsement. At least that someone can be made to pay the cost of trust, which makes such collusion difficult to begin with.

Colluded Empty Promise

There may be a colluded empty promise by W_0 and W_n so that W_{n-1} is victimized. This means that W_0 escapes too, in which case W_1 can take over the responsibility of the drawer. If it fails and the responsibility is forwarded up to W_{n-1} , it is indistinguishable from the state in which every W_i where $0 \leq i < n - 1$ is colluding. The rest is the same as the case of a colluded defaults.

Entrapments

Another form of colluding may be to entrap W_i so that it looks as if W_i committed a misbehavior such as an empty promise. This is only possible with a compromised secret key or a forged key pair, because there needs to be a verifiable signed message to prove that W_i did it. This requires a breach of the trust model.

4.8 Distributed Auditing

Definition 25 (estimated trust) *An estimated trust of a participant, or their likelihoods not to default, is a function of their debt, history of redemption, credit and history of usage.*

The actual estimation function to be used is selected depending on the participant's experiences. They may want to apply Bayesian inference, for example.

Since i -WAT is decentralized, estimation of the trust of other participants needs to be achieved by collecting information from each trade, and constructing an image of the participant's current balance and history based on that information. There is no guarantee that the collected information is truthful if there are incentives for lying or colluding.

In order to tackle this problem, the author first takes a look at how a participant's balance and history information can be used for estimating their trust. Table 4.4 shows how records of tickets in one's checkbook can be used in measuring their trust.

Table 4.4: Meanings of tickets in the *i*-WAT book of a user

<i>Drawer</i>	<i>Description</i>	<i>Meaning</i>
This user	Redeemed	Balanced income and outlay
This user	Not redeemed	Negative component or debt
Other user	Possessed	Positive component or credit
Other user	Used	Balanced income and outlay

The author argues that the following statements are true:

1. There is no incentive to conceal the records of redeemed or used tickets.

Participants would not claim less income and outlay in balance than there actually is because it is likely that it would decrease their estimated trust.

2. One cannot lie to have more debt by claiming to have drawn more tickets than they actually have, or more credit by claiming to possess more tickets than they have, because they may be asked for proofs in auditing processes.

If there are proofs for transactions which never happened, the author believes that it should be considered a set of different problems: transactions without actual practice of bartering, and inflation in the system of trust which might result from such transactions. The author believes there can be operational solutions for this sort of problems, and has experimented on a solution as described in sections 5.3.3 and 6.2.1.

3. The only reasonable way to tell a lie is not to reveal the existence of debts or credits.

As a countermeasure for this, we can apply the protocol for *fair sharing* described in [61].

Protocol to detect concealed debts (CD):

CD-1. At a random interval, to a randomly chosen user, one asks for a list of their possession of tickets.

CD-2. For each ticket in the list, that one asks its drawer for the list of their drawn tickets.

CD-3. If the ticket in question is not included in the list, the drawer is lying about their debts.

Protocol to detect concealed credits (CC):

CC-1. At a random interval, to a randomly chosen drawer, one asks for a list of their drawn tickets.

CC-2. For each ticket in the list, that one asks its current owner for the list of all tickets they possess.

CC-3. If the ticket in question is not included in the list, the owner is lying about their credit.

Note that in the above protocols, CD-1 and CC-2, as well as CD-2 and CC-1, are indistinguishable to the receiving end of the queries. Therefore there are disincentives to lie to the queries.

A drawer and the lender may have a reason to collude. They might lie that the transaction never took place. However, the relationship between a drawer and the lender is not symmetrical. It is riskier for the lender because lying means denial of their crediting debt.

4.9 Public Key Exchange

4.9.1 Overview

PGP has historically used key servers to distribute public keys.

However, because of privacy reasons, the author argues that use of key servers is no longer recommended; a public key is associated with user IDs which are conventionally e-mail addresses of the user, and a key can be associated with photo IDs which are typically portraits of the user. Placing a public key on key servers allows these information publicly accessible, which may lead to unwanted consequences such as spamming[107] and stalking.

Moreover, depending on key servers decreases the level of achieved autonomy.

Integration of *i*-WAT is to provide means to distribute public keys in a P2P way.

4.9.2 Propagation of Signatures

To achieve this, the Transport Overlay Layer is further modularized to accept to convey public keys as well as *i*-WAT tickets. Satisfaction of the conceptual reliable multicast by the communication channels lets public keys propagate over a group, whose authenticities are verified by each party by way of fingerprints.

The exchange core defines an XML data structure for public keys.

4.9.3 Support for the preconditional properties

The reference implementation lets users exchange their public keys directly without having to consult a public key server, as described above.

From a user's point of view, this is performed by choosing a correspondent from a buddy list of a presence sharing system, and selecting either importing or exporting their keys. When imported, a window pops up with the fingerprint of the public key, asking the user whether to sign the key or not. This is expected to enhance the ease for *mutual signing by knowing*.

The reference implementation currently does not directly support *mutual signing by participation*. However, its current design uses the buddy list to locate the owner of a public key, so that new participants will be required to add the drawer in their buddy list if they have not already, to which the drawer would respond by adding them back. Then the above mechanism for key exchange can be used.

The publicly available version of the reference implementation currently does not have a support for *mutual full trust by participation*, but it has been experimented as described in sections 5.3.6 and 6.2.3.

4.10 Extension Mechanism

4.10.1 Currency Name Space

The semantics of an XML data structure is defined in the domain of an XML name space. Likewise, the data structure for *i*-WAT tickets allows the designers to specify the name space for a specific currency, which defines the semantics of the currency.

Figure 4.11 shows an example of the data for an *i*-WAT ticket. In the figure, *xmlns* attribute in `<x/>` element specifies an XML name space, indicating that the data is in the context of *i*-WAT. The `<sum/>` element within has *ns* attribute which specifies the name space for a currency. The name space is given typically by a URL, so that it can be autonomously specified under the condition that the domain name has already been registered.

4.10.2 Currency Semantics

The semantics of a currency is further to be defined by a software, which may handle such properties as the region, group, duration in which the tickets are usable and so on.

The reference implementation allows a designer to define the semantics of a new currency by a description of an XML element, in case the basic semantics of the currency is no different from the default *i*-WAT currency. Figure 4.12 shows an example of the definition for an extended *i*-WAT currency.

```

<x xmlns="http://www.media-art-online.org/iwat/">
  <signed>
    <draw creditor="bob@media-art-online.org"
      debtor="alice@media-art-online.org" id="1462394433">
      <sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
      <min>0</min>
      <var per="week">
        <constant value="-0.0010"></constant>
      </var>
      <memo>I will draw a dog for you.</memo>
    </draw>
  </signed>
  <signature>
    iD8DBQBduHu9dz1H60eon3cRAntXAJ49dSG4lhEjcXqLfhtUtVMc2xxSuACeI5I7HKiy
    LXswsaHHcCLv4DKAONQ=
  </signature>
</x>

```

Figure 4.11: An example of an *i*-WAT ticket data

```

<?xml version="1.0" encoding="UTF-8"?>
<iwat-plugin>
  <name lang="en" default="yes">WIDE Hours</name>
  <ns default="yes" variance="yes">http://fran.sfc.wide.ad.jp</ns>
  <unit>hours</unit>
  <statements lang="en" default="yes">WIDE Hours:
    The drawer promises to pay by a commodity or service that is agreed
    with the user, when a WIDE member or their associate claims for
    repayment by presenting this ticket.
    This ticket is accepted as means of payment among WIDE members and
    their associates.

    WIDE Project pursues issues and problems concerning construction of
    the distributed system which connects all computers on the planet,
    and contributes to well-beings of people and the societies.
  </statements>
  <default-value>1:00</default-value>
  <min-value>0:05</min-value>
  <max-value>24:00</max-value>
  <banner>images/hours-banner.png</banner>
</iwat-plugin>

```

Figure 4.12: An example of an extension to *i*-WAT currency

4.11 Coexistence with Existing Currencies

This section describes the inter-networking with other currency systems as discussed in [75].

4.11.1 Exchange Points and Translation Mechanism

The semantics of *i*-WAT inherited from the WAT System allows the tickets to be freely associated with values. Such values include *i*-WAT tickets in different units in different currencies.

Figure 4.13 shows how *i*-WAT tickets in different currencies can be exchanged with one another.

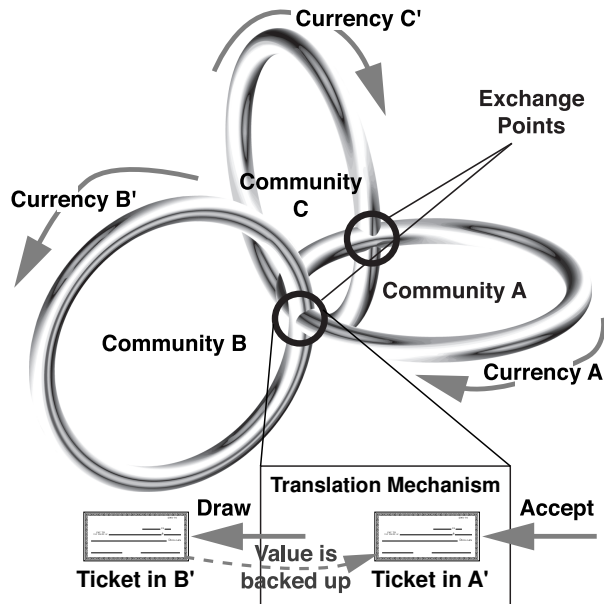


Figure 4.13: Exchanging *i*-WAT tickets among different currencies

The figure shows three communities, *A*, *B* and *C*, depicted as rings. These communities can be circles of people, rings of Chord[90] (or some other forms of distributed hash tables), or just any groups of nodes in autonomous overlay networks. It is assumed that methods for message-routing exists among these communities. The communities use currencies *A'*, *B'* and *C'* respectively.

An entity belonging to both communities *A* and *B* can become an exchange point between currencies *A'* and *B'*. Such an exchange point can take a ticket in *A'*, and draw a new ticket in *B'*, or vice versa. The value of the new ticket is backed up by the exchange point's possession of the original ticket.

A participant in community A can ask the exchange point to draw a ticket in B' in return of a ticket in A' . The obtained ticket can be used to ask for some service in community B . i -WAT requires that the each end of a transaction must have the other's validated public key. Those public keys can be signed by the exchange point.

The exchange points are motivated to collect the drawn tickets and give up the original tickets, as it is likely that it will insure the increase of their estimated trust. They are also motivated to advertise their services.

If someone in community B wants to issue a ticket in currency C' , then they use the two exchange points in Figure 4.13 to exchange a ticket in B' to A' and A' to C' .

4.11.2 Internetworking with an MCS

i -WAT can also interconnect with a currency based on an MCS, which has been experimented as described in sections 5.3.3 and 6.2.1. Figure 4.14 illustrates how it works with an example of a currency called WIDE Hours, a member's currency for WIDE Project[101].

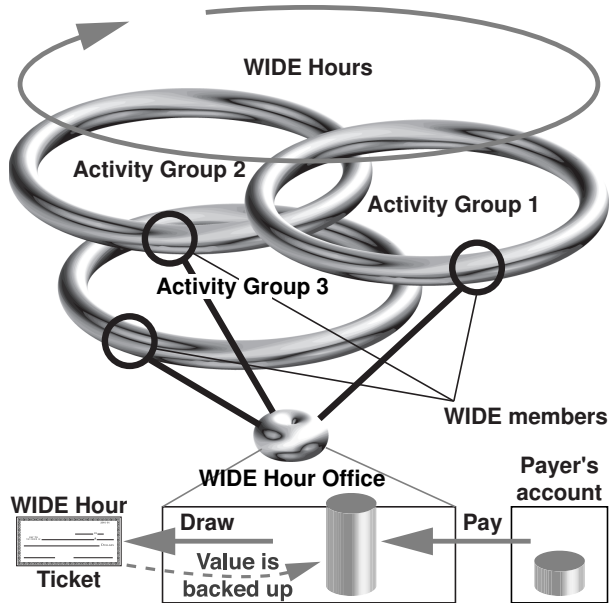


Figure 4.14: Exchanging MCS-based WIDE Hours outside its members

WIDE Project has a strict notion of membership, but its activities often involve non-members. While it does not always make sense to provide non-members with accounts in WIDE Hours, i -WAT tickets having the unit of WIDE Hours can always be issued outside the project. By an internetworking mechanism, such tickets can be made compatible with the MCS version

of WIDE Hours.

In the figure, there are two types of overlay networks. One is the overlays of activity groups, and the other is the overlay of MCS version of WIDE Hours. The former is P2P and the latter is a star network. The former overlays can interact with each other using *i*-WAT tickets in the unit of WIDE Hours. A WIDE member can obtain such tickets by asking WIDE Hours Office for exchanges.

It works just like the exchange point in Figure 4.13, only that the office accepts payments in MCS, which backs up the value of the new *i*-WAT ticket.

4.12 New Economic Order (NEO)

As an example of a specific design of an autonomous, distributed collaborative system, the author describes the concept of NEO (New Economic Order), which is a new way to treat information and their providers in the economy of digital networks.

Information has a near-infinite productivity once it is formed, because it can be copied at a minimum cost. Treating information as commodities together with physical resources and labors, including computing power, storage space and communication channels, is potentially problematic, because information providers can sell their information virtually an infinite times, accumulating purchasing power not comparable to those of others.

In NEO, participants agree that information is shared for free, but the providers of information are supported by their allowance to issue *reduction* tickets.

This is compatible with the GNU manifesto[25] which states that software must be freely shared while programmers need support from communities. NEO is potentially useful for building economy around a free or open source software.

NEO is simulated in section 6.1.6.

Chapter 5

Practice

5.1 Reference Implementation

5.1.1 Overview

i-WAT allows the underlying conveyer of messages to be existing e-mail or instant messaging systems. As the reference implementation, the author has developed an *i*-WAT plug-in and the hosting Jabber/XMPP client called *wija*. The software is available from <http://www.media-art-online.org/wija/> (the *i*-WAT plug-in is bundled with all platform-specific packages).

i-WAT, as well as public key exchange to support the system, have been implemented as extensions to the Jabber instant messaging protocol.

There is also a more experimental implementation of *i*-WAT over a wireless ad-hoc network; the author has developed a CCS client called *wijapo* (for *wija* pocket), and ported the *i*-WAT plug-in onto it.

For conducting experiments, the author has also developed an MCS toolkit called OMELETS (Open, Modular and Extensible LETS).

5.1.2 Outline of Jabber/XMPP

History

Jabber/XMPP[69, 70] is a set of open protocols for instant messaging and presence sharing.

It was first invented by Jeremie Miller[103] in 1998. The project was made open to public the next year, and an open source Jabber server project *jabberd* began, whose first version was released in year 2000. JSF[42] (Jabber Software Foundation) was established in 2001 to maintain the specifications and to support their enhancements. In 2004, the core protocols of Jabber were standardized by IETF[95] as Extensible Messaging and Presence Protocol (hence the name Jabber/XMPP).

Jabber/XMPP has been applied to many instant messaging systems including those developed by Apple, FedEx, Google, Oracle and Sun Microsystems.

Characteristics

The characteristics of Jabber/XMPP in comparison with other instant messaging systems are as follows:

1. *Openness*

Specifications of Jabber/XMPP are open to public. Anyone can implement the protocols for free. Readers can find lists of clients and servers at [42].

2. *Autonomy*

The network of Jabber/XMPP, as illustrated in Figure 5.1, is close to that of SMTP; mail servers can be set up by anyone without permissions from any parties once a domain name is registered, and anyone can freely join the network of the global electronic mail system. Likewise, XMPP servers can be set up without any permissions, and anyone can freely join the global network of the instant messaging and presence sharing system.

3. *Extensibility*

Jabber/XMPP is an extensible set of protocols whose data descriptions are based on XML. New features can be incorporated without breaking the existing system. Extended protocols can be submitted to JSF as a JEP (Jabber Enhancement Proposal) for standardization. A JEP is approved as a standard after feedbacks from implementers and reviews from JSF.

These openness, autonomy and extensibility are also basic properties of the Internet. IP is an open protocol so that anyone can implement routers or protocol stacks that runs on various computers, and we are free to connect networks at homes, schools or corporations to the Internet, and to become part of it. Popular protocols such as HTTP[7, 23] (Hypertext Transfer Protocol) and SMTP are seen as extensions of IP. As it is eminent in the case of P2P technologies, new services keep emerging as new protocols are proposed and tested.

The author argues that Jabber/XMPP is like another Internet over IP. Jabber/XMPP is just one of extensions of IP as HTTP and SMTP are, but it is designed in such a way that a new world of communication can emerge from it.

Still, Jabber/XMPP is a server-based technology so that it is the author's intention that it will be replaced by a more P2P-oriented technology in the

future, to achieve the level of required autonomy on the Internet as well as on wireless ad-hoc networks.

Communication Mechanism

In Jabber/XMPP, a peer is identified by a Jabber ID in the following form:

```
user@domain/resource
```

A *user* name defines a user in a domain. A *domain* name is effectively the host name of the server to which the user is connected with a client. A *resource* name distinguishes a communication session from others by the same user at the same domain.

Figure 5.1 illustrates the communication mechanism of Jabber/XMPP.

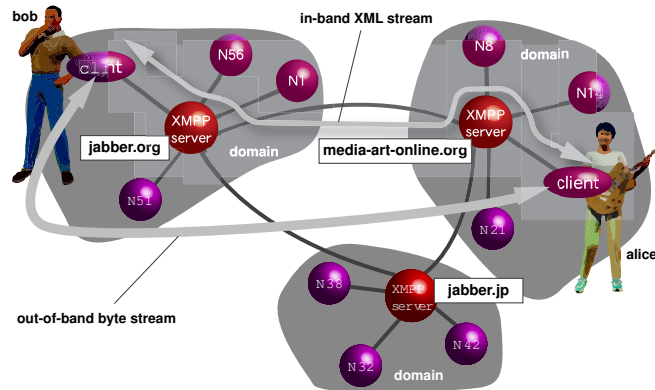


Figure 5.1: Overview of communication in Jabber/XMPP

Like SMTP, clients need to talk to a server in order to have their messages reach the clients on the other ends (*in-band XML stream*). It implies that Jabber/XMPP has such a weakness that communication becomes impossible when one of servers in between is down.

Jabber/XMPP also provides means for the clients to directly communicate (*out-of-band byte stream*), using protocols such as SOCKS5[45]. It is recommended that clients use out-of-band communication whenever a large data is involved. Still, in-band communication takes an important role in identifying the peers by their IP addresses and port numbers, which can dynamically change; because of this, Jabber/XMPP provides an excellent way to rendezvous for applications on the Internet.

5.1.3 *wija*

wija is a Jabber/XMPP client developed for implementing *i-WAT*. It is a free software licensed under GNU GPL[24]. *wija* is mostly written in Java,

and runs on many operating system platforms such as Linux[47], Mac OS X[4] and Windows[50].

Figure 5.2 shows some screenshots of *wija* and its plug-ins.

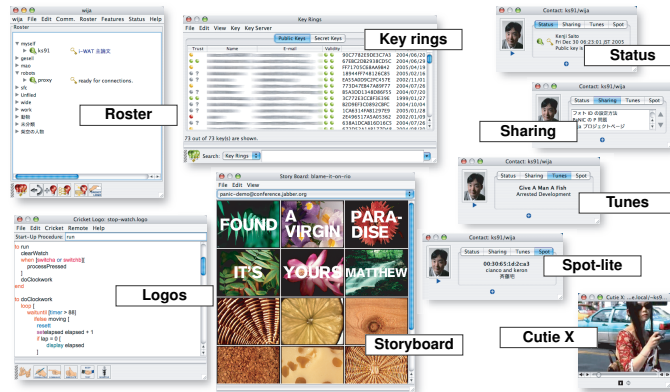


Figure 5.2: Screenshots of *wija* and its plug-ins

Table 5.1 shows the list of plug-ins for *wija* bundled in the official distribution, including the reference implementation of *i-WAT*. Table 5.2 shows the list of JEPs (partially) implemented in *wija* and its bundled plug-ins.

Table 5.1: Built-in plug-ins for *wija*

Name	Description
<i>i-WAT</i>	The reference implementation of <i>i-WAT</i>
Tunes	Shares the tunes played by Apple Computer's iTunes software on <i>wija</i> -activated computers.
Spot-lite	Shares the user's physical locations by the BSSIDs (Basic Service Set Identifier) of their connected wireless networks.
Logos	A programming environment for Logo language and MIT's Cricket computers.
PaNIC	Collaboration support by storyboards, etc.
Cutie X	An interface with Apple Computer's QuickTime.

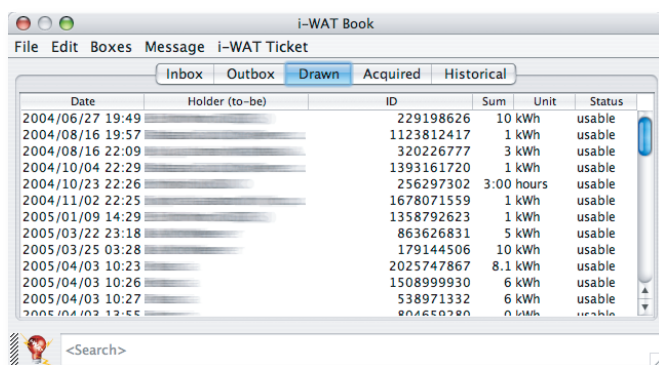
wija is end-to-end oriented, which means that problems in a new application are not to be solved by adding new features to the servers, but by communication among clients. This makes safe communication among clients more important, in addition to the necessity imposed by the design of *i-WAT*.

wija provides encrypted and signed communication over Jabber/XMPP using OpenPGP (currently assuming GnuPG[93] as its implementation), as well as features to handle rings of public keys, their validities and the owners' trust.

Table 5.2: JEPs implemented in *wija* and its bundled plug-ins

<i>JEP Number</i>	<i>Name</i>
JEP-0020	Feature Negotiation[51]
JEP-0027	Current Jabber OpenPGP Usage[55]
JEP-0030	Service Discovery[35]
JEP-0045	Multi-User Chat[73]
JEP-0047	In-Band Bytestreams[43]
JEP-0065	SOCKS5 Bytestreams[88]
JEP-0079	Advanced Message Processing[52]
JEP-0082	Jabber Date and Time Profiles[68]
JEP-0086	Error Condition Mappings[62]
JEP-0095	Stream Initiation[56]
JEP-0096	File Transfer[57]
JEP-0112	User Physical Location[71]
JEP-0115	Entity Capabilities[36]
JEP-0118	User Tune[72]

Figure 5.3 shows a screenshot of the *i-WAT* book, which is the software component to manage *i-WAT* messages and tickets, at part of the *i-WAT* plug-in for *wija*.



Date	Holder (to-be)	ID	Sum	Unit	Status
2004/06/27 19:49		229198626	10 kWh	usable	
2004/08/16 19:57		1123812417	1 kWh	usable	
2004/08/16 22:09		320226777	3 kWh	usable	
2004/10/04 22:29		1393161720	1 kWh	usable	
2004/10/23 22:26		256297302	3:00 hours	usable	
2004/11/02 22:25		1678071559	1 kWh	usable	
2005/01/09 14:29		1358792623	1 kWh	usable	
2005/03/22 23:18		863626831	5 kWh	usable	
2005/03/25 03:28		179144506	10 kWh	usable	
2005/04/03 10:23		2025747867	8.1 kWh	usable	
2005/04/03 10:26		1508999930	6 kWh	usable	
2005/04/03 10:27		538971332	6 kWh	usable	
2005/04/03 12:55		806650280	0 kWh	usable	

Figure 5.3: *i-WAT* book

5.1.4 *wijapo*

wijapo is a more experimental sibling of *wija* that runs on a handheld device. The current version is developed for a Pocket PC device[105].

It is designed as a CCS[41] client, and provides a primitive to communicate over a series of link-local broadcasts (either IPv4 or IPv6, depending

on the preferences¹) to convey messages described in XML in a probabilistic manner.

5.1.5 OMELETS

The author has developed OMELETS[75], a collection of Java classes to implement an MCS as a web application, in the hope that it becomes useful in verifying the designs of mechanisms using barter currencies.

Among applications of OMELETS are WIDE Hours[100] and MANA[59], which will be explained later in section 5.3.

5.1.6 Public Relations

The author has been advertising *wija* and *i-WAT* via a number of media.

Figure 5.4 shows the top web pages of *wija* and *i-WAT*. A Wiki site[32] for *wija* has also been made open to public.

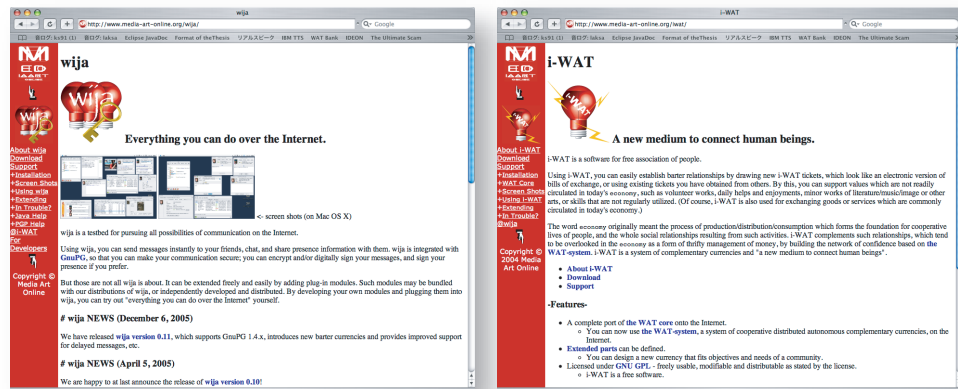


Figure 5.4: *wija* (left) and *i-WAT* (right) top pages

wija is among the lists of Jabber/XMPP clients at the JSF web site and an entry in Wikipedia[104]. The author has introduced *wija* in a special article in the January 2006 issue of the monthly JavaWorld in Japan[77].

As for more community oriented activities, the author has set up community pages for *wija* at SNS (Social Networking Service) sites *mixi*[20] and *GREE*[30]. There is a *wija* developers mailing list for discussions.

¹J9[38], a Java VM chosen for implementing *wijapo*, supported IPv4 only as of July 2005.

5.2 Implementation Issues

5.2.1 Implementation of *wija*

Interfacing with GnuPG

The functionalities of GnuPG are accessed from *wija* by calling *gpg* command through *exec()* method of the Java runtime module. The functionalities are encapsulated as a Java class named *org.media_art_online.gnupg.GnuPG*, so that plug-in writers need not to be bothered by the detail of using GnuPG. The package *org.media_art_online.gnupg* provides Java classes to abstract GnuPG data structures such as keys, user IDs and photo IDs.

Linkage with Plug-ins

There was a challenge of providing a cross-platform mechanism to link plug-ins with *wija*. This is done by class loaders provided by the Java virtual machines.

This mechanism requires the file name of the JAR (Java ARchive) file of a plug-in to be named as follows:

```
full-name-of-the-class.jar
```

For example, the plug-in JAR file for *i-WAT* is named as follows:

```
org.media_art_online.iwat.Iwat.jar
```

Upon start-up, *wija* searches for those files under the *plugins* directory inside the program directory. When it finds one, it tries to load the class specified by the file name from the JAR file, which will result in loading all available referred classes recursively. All classes of found plug-ins are loaded first, and then the initialization code of each plug-in is called. This allows a plug-in to call public methods and to access public fields of other plug-ins.

This mechanism somehow does not work under Windows in some conditions if an older version of Java 2 runtime environment (earlier than J2SE 5.0) is used.

Hypertext Sharing

Hypertext sharing provides an illusion of direct retrieval of data using a web browser from a client at the other end through a web server running on their machines, even though both computers can be inside their own private networks or within firewalls.

This is realized by running a pseudo-HTTP server locally inside *wija*, and having the web browsers of the user's choice access the server on the localhost. The server initiates streams for file transfers, via a proxy service if necessary, and all data transfers are performed at the back as in-band or SOCKS5 byte streams.

Proxy Discovery

All *wija* clients are designed to be capable of providing a proxy service. This feature can be turned on and off by the users.

When a direct SOCKS5 connection is found impossible, *wija* searches for an available proxy service in its buddy list. This allows users to set up a proxy service on some machine using an only regular distribution of *wija*², and make the service available to them by just adding the node onto their buddy lists.

There is a proxy service provided by the author; its Jabber ID is as follows:

proxy@media-art-online.org

5.2.2 Implementation of *wijapo*

Choice of the Java Virtual Machine

The author discovered that capabilities of the currently available Java virtual machines for the Pocket PC were limited, as to porting the mechanisms of *wija* as described in the former sections onto the platform.

The author found that J9[38] was one of the best available virtual machine, for its performance and features such as a support for JNI (Java Native Interface). JNI was especially important because lack of features could be complemented by writing a native modules.

Choice of the GUI Toolkit

It turns out that Swing[91], a standard toolkit used for implementing the GUI of *wija*, was unavailable on Java virtual machines for Pocket PCs. Which necessitated the author to use SWT[92] instead of Swing to implement the GUI of *wijapo*. Since SWT is designed to be lightweight, its interaction model is different from that of Swing, which made it costly in terms of time to redesign and implement *wijapo* and the *i*-WAT plug-in for it.

To make matters worse, the code of the package *org.media_art_online.gnupg* is in some part dependent on the underlying graphic model of the system, because of its need to handle photo IDs. The differences between Swing and SWT turned out to be problematic in keeping the code stable, and *wijapo* is currently removed from the code base of *wija* family (it is to be reimplemented in the future).

²Although *wijabot*, a sibling of *wija* without graphical user interface and with supports for automation, is also available.

Data Integrity

The message content to be sent via the mechanism of CCS is tagged with its hash value generated by SHA-1[60]. Upon receipt, their integrity is checked by comparing the hash value with the one calculated from the data. If unmatched, the data is silently discarded.

5.2.3 Implementation of *i*-WAT

Detection of Double-Spending

Detection of double-spending is automated by the reference implementation. The drawer's software checks whether or not the outer-most inner data structure of an <accept/> message which has <draw/> or <use/> element equals the corresponding data in their *i*-WAT book. If it does not, the software decides either it is double-spent or the message is spurious, and suggests the user to disapprove the transaction (it does not allow them to approve it).

Disapproval is not yet automated in the reference implementation, and the user necessarily intervene to sign their messages for either approving or disapproving transactions. This is because it is debatable whether or not it is safe to allow the software to hold passphrases for some time period for automatic signatures. The author believes it is a trade off between usability and security, and for the time being, the stress is more on the security side than usability.

The Cost of Verification

The drawer's software verifies the outer most signature of an <accept/> message, and then compares the signed content; this allows verification time to remain approximately constant as the chain of endorsements grows.

Implementing the Security Rule

Receivers' software searches for <draw/> element in the <use/> message to discover the drawer of the ticket, which it assumes to find at most once. This algorithm can be modified to search for the *outer-most* <draw/> element to allow the role of the drawer to be switched in case the security rule has to be applied.

This feature has not been implemented in the reference implementation as of January 2006.

Implementing Variance Over Time

This feature is relatively new, and has been introduced since April 5, 2005.

Data Structure *Reduction* and *multiplication* tickets were incorporated to the system with minimal changes to the message format of the existing *i*-WAT protocol.

In the message format, an *i*-WAT ticket is represented by an XML data in which the value is expressed in the following element:

```
<sum ns="name-space-URL">value</sum>
```

where *name-space-URL* specifies the currency unit in which the debt is quantified. Variance over time is implemented by adding XML elements for the minimum/maximum value and the variance rate.

The ROT feature, for example, is implemented by adding the following set of elements to the XML data:

```
<min>minimum-value</min>
<var per="time-unit">reduction-rate</var>
```

Currently, constant value only is supported as an expression in place for a *reduction-rate*:

```
<constant value="constant-value" />
```

For compatibility reasons, *name-space-URL* is accompanied by a trailing `#var` in case of a *reduction* or *multiplication* ticket. This way, older implementations of *i*-WAT can assume that the ticket is of unknown unit (because the name space string does not match with the ones they support), avoiding situations where the effective value of a ticket is wrongfully presented to the users.

Agreement on Time Since it is unrealistic to assume that the clocks of all participating computers are synchronized with precision, the time is ultimately measured by the computer of the drawer.

The timestamp of the drawer's or user's signature (depending on whether the trade is issuing or circulation) defines the effective value of a *reduction* or *multiplication* ticket, to which the recipient either agrees or disagrees. In this sense, time is measured by the computers of receivers, and the last receiver is necessarily the drawer (or the one assuming the role of it). The drawer (or their software agent) is responsible to check that the timestamp belongs to the past for them when they approve a transaction in order to avoid unwanted increase or decrease of the value.

5.3 Experiments

5.3.1 Overview

Table 5.3 is the chronological list of experiments related to *i*-WAT. *i*-WAT was preliminarily introduced in March 2003, followed by a series of experiments using MCS implementations. *i*-WAT was introduced to the public

in June 2004. The Vegetable Trading experiment in July 2005 was the first appearance of *i*-WAT-equipped wireless handheld devices.

Table 5.3: Chronological list of experiments

<i>Period</i>	<i>Experiment</i>
March 2003	Preliminary deployment of <i>wija</i>
September 2003	MCS-based WIDE Hours
October 2003	MCS-based MANA
March 2004	WIDE Hours interconnection
June 2004	Public release of <i>wija</i>
July 2005	Vegetable Trading

5.3.2 Preliminary Deployment of *i*-WAT

i-WAT and *wija* were preliminarily introduced to WIDE members in March 2003. Although *wija* was welcomed as a new implementation of a Jabber/XMPP client, the author found that only a few WIDE members (7 out of 264 participants in a 4-day meeting) were enthusiastic enough to try *i*-WAT trading.

5.3.3 WIDE Hours

Overview

Assuming that distributed auditing described in section 4.8 is implemented, we can approximate the P2P barter currency by a centralized model (we can only approximate it because the results of distributed auditing will be probabilistic).

An application of OMELETS have been developed for WIDE Project to implement their barter currency called WIDE Hours[100] (the web site is for WIDE members only), whose values are expressed as the hours of work for the project.

WIDE Power, an experimental trust value to estimate qualities of traders, is introduced. It is given by the following formula:

$$WIDE\ Power = \log \frac{income \times outlay}{|income - outlay| + 1} - penalty$$

where *penalty* is decided by the administration.

The maximum WIDE Hours each member can spend is limited to 24 WIDE Hours a day.

September 2003 Experiment

WIDE Hours was first introduced in a four-day meeting of WIDE Project in September 2003. The purpose of the experiment was to understand behaviors of participants in such a case that increasing the estimated quality as traders works as an incentive to participate. There was a ranking page on the web site (left of Figure 5.5) to give feedbacks to the participants.

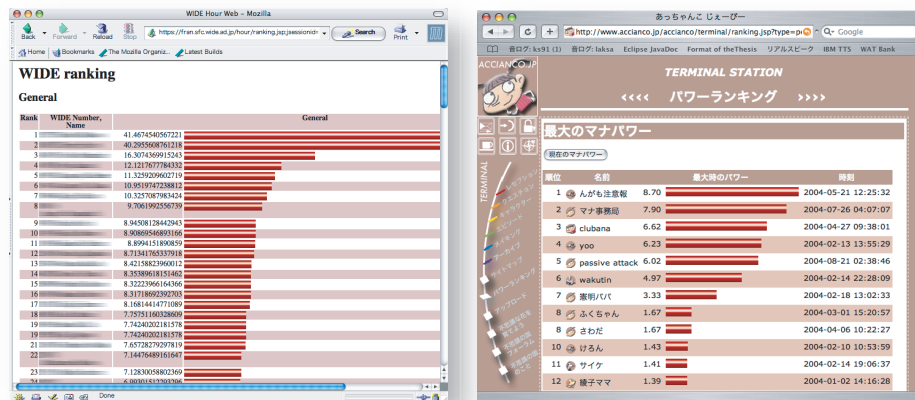


Figure 5.5: WIDE Hours (left) and MANA (right) ranking pages

More than 500 transactions were processed in one month after the debut of WIDE Hours, but it turned out that most transactions were just to improve their ranks.

March 2004 Experiment

In March 2004, the MCS version of WIDE Hours was interconnected with its *i*-WAT counterpart (Figure 5.6), as described in section 4.11.2.

Also, the calculation of ranking was improved by introducing some additional metrics than just a balance and history, such as variety of trade partners, and the length of the chain of endorsements in case of the *i*-WAT version of the currency.

5.3.4 MANA

MANA is another application of OMELETS, involving a book[58] equipped with an Auto-ID[5]-compliant 2.45GHz RFID (Radio Frequency Identification) tag.

The barter economy of MANA allows the users to obtain points by visiting certain locations where RFID readers are placed, and having their books

Figure 5.6: Example: *i*-WAT version of *WIDE Hours* in circulation

identified by the readers. Obtained points can be used in a community residing on the web[59] (right of Figure 5.5).

5.3.5 Public Releases of *wija*

Table 5.4 is the chronological list of releases of *wija*. Statistical analysis on the publicity of the latest version (0.11) is found in section 6.3.2.

Table 5.4: Chronological list of *wija* releases

<i>Date</i>	<i>Version</i>	<i>Description</i>
Jun 14, 2004	version 0.03	Pre-public release version.
Jun 14, 2004	version 0.04	Initial public release.
Jun 21, 2004	version 0.05	Improved data exchange.
Jul 15, 2004	version 0.06	Many improvements for usability.
Sep 29, 2004	version 0.07	Key rings, photo IDs and Spot-lite.
Jan 3, 2005	version 0.08	Hypertext sharing and Tunes.
Feb 2, 2005	version 0.09	Many improvements for usability.
Apr 5, 2005	version 0.10	Variance over time and PaNIC storyboard.
Dec 6, 2005	version 0.11	GnuPG 1.4.x support and new barter currencies.

5.3.6 Vegetable Trading

Overview

The author has experimented on *i*-WAT trades and PGP public key exchanges by a group of wireless handheld devices communicating in an ad-hoc manner, together with a team of CCS-experimenters. The experiment was conducted at EXPO 2005 AICHI JAPAN.

The main purpose of the experiment was a proof of concept that *i*-WAT can be used over wireless ad-hoc channels of communication. Which is one step forward for *i*-WAT toward becoming a tool for mutual help under such situations like post disasters (ex. earthquakes), where energy and connectivity are insufficient.

The Game

1. Participants (30 volunteers) are divided into five vegetable groups (carrot, onion, potato, green pepper and eggplant) and one office group. Each group consists of five people.
2. A member of a vegetable group starts with one vegetable at hand. For example, a member of the *carrot group* has a carrot when the game begins.
3. A vegetable can be traded with an *i*-WAT ticket.
4. When a member of a vegetable group had two different kinds of vegetables, he or she reports to an office to get one point. They can obtain just one point from the same combination of two vegetables. Therefore, they can obtain at most 10 such points.
5. When an *i*-WAT ticket returns to the drawer, they report to an office so that everyone in the ticket's chain of endorsements obtain the points proportional to the length of the chain.
6. In the end, the group and the individual who obtained the highest points are awarded.

Figure 5.7 shows how issuing is performed in the game. It involves a physical contact and a public key exchange prior to drawing an *i*-WAT ticket.

Figure 5.8 shows how circulation is performed in the game. Likewise, public key exchange needs to be performed prior to the use of the *i*-WAT ticket, but it is more challenging as the drawer may have moved to a different place, who needs to approve the transaction (although CCS is capable of multi-hop communication, this experiment was conducted in a small area where everyone is reachable by a direct wireless communication).

The Metaphors

The game is designed to be metaphorical to a post-catastrophic situation where the social infrastructures are broken.

A member of a vegetable group is a person in such a situation.

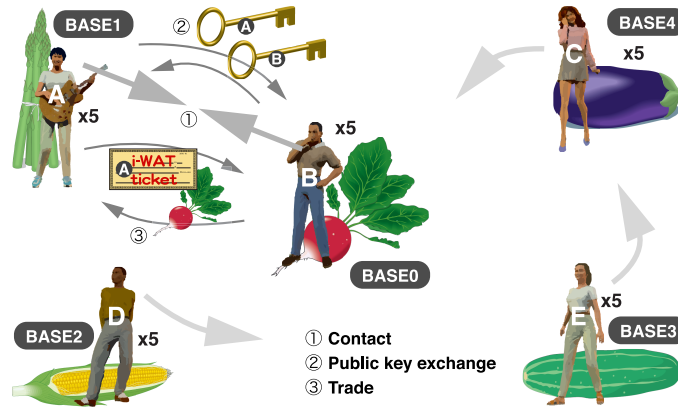


Figure 5.7: Issuing in Vegetable Trading

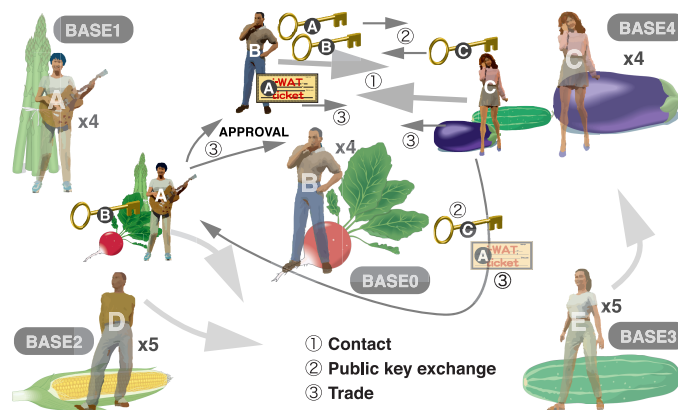


Figure 5.8: Circulation in Vegetable Trading

A **vegetable** is something necessary for the survival of people. Everyone has something to provide.

A **point obtained from a combination of vegetables** denotes that necessary goods or service was obtained.

A **point by redemption** means that tickets with longer chains are more trustworthy.

Offices are placed for collecting data, and are unrelated to actual situations.

The behavior of participants to obtain higher points corresponds to the principle of action in actual situations, in which people want to secure their lives and take smaller risks.

Fingerprint Checking

Since the game starts by assuming no trust on the participant's public keys, checking fingerprints and signing keys of others take an important role in the game, as they do in the actual operation of *i*-WAT.

This checking was made easy by translating the fingerprints into color patterns (Figure 5.9); participants just need to see the pattern shown on the screen of the partner's terminal, and compare it with the pattern shown on their terminals.



* Actual screens were formatted according to the dimension of the terminal screen.

Figure 5.9: Main (left) and signing (right) screens of Vegetable Trading

Automated Trust Settings

In this experiment, trust of the public key owners are automatically set when a participant had a successful trade with them (implementation of the *mutual full trust by participation* in Property 7).

Photo ID Shooting

The device used in the experiment was equipped with a digital camera. Public key photo IDs of the participants were registered by shooting their own portraits with the camera before the game started. This was useful not only for the participants themselves to look for their trade partners, but for the analysis of the chains of endorsements of resulted *i*-WAT tickets.

5.4 Simulation

5.4.1 Principles

This simulation is to help understanding the inherent properties of currency systems by using the simplest model possible. Thus, in the model, there is no money bill, there is only one bank and no financial market. This is a model of a pure trust economy.

The bank may behave as an MCS by setting the interest rate to zero.

This model and a comparative study of risks among complementary currencies were first introduced in [82].

5.4.2 The World

The world consists of a set of participants U such that $|U| = 2500$ (for simulation scenarios with *regular* tickets only) or $|U| = 500$ (for simulation scenarios with *reduction* and *multiplication* tickets), set of materials (of equal values) M such that $|M| = 100$, and a manufacturer function f such that $f : U \mapsto M$. The manufacturers are evenly distributed, and approximately equal number of participants $u \in U$ manufacture a material $m \in M$, which they can trade with others.

The participant forms a network $\langle U, K \rangle$ where K is an acquaintance relation (also called *links*) such that $K \subset U \times U$. It is assumed that the relation is symmetric: xKy always implies yKx . Initially, the population forms a scale-free network, where smaller number of participants know greater number of them, and larger number of them are acquainted with smaller number of them. All participants can be reached from any participants in a small number of hops. Figure 5.10 and TABLE 5.5 show the properties of the initial networks.

Table 5.5: Initial properties of the small worlds

Property	Population		
	2500	500	100
Number of unreachable pairs	0	0	0
Mean distance (in hops)	4.18	3.46	2.71
Maximum distance (in hops)	7	6	5

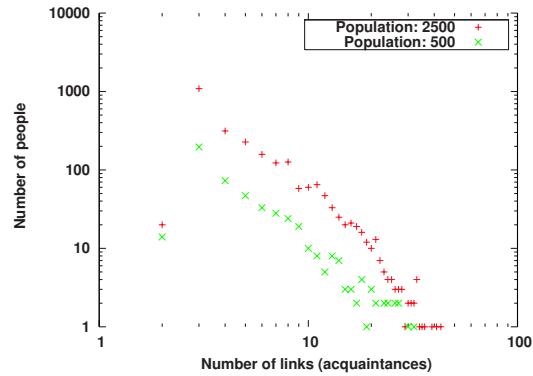


Figure 5.10: Link distributions in the small worlds

Population = 100 is a sample case. Figure 5.11 is a visual representation of the sample network.

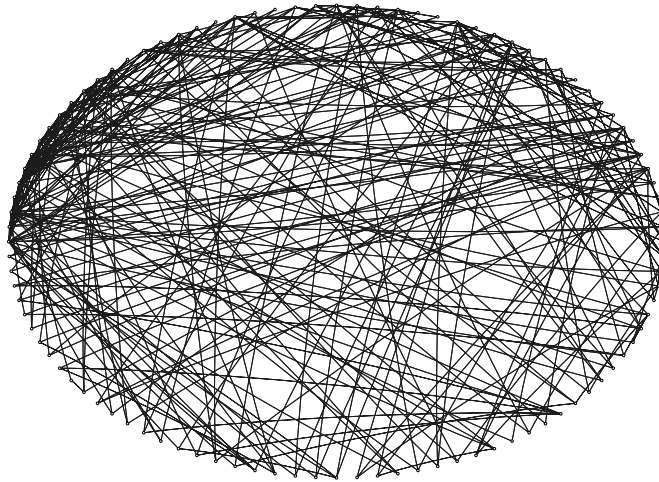


Figure 5.11: Sample initial world of population = 100

Time is abstracted as a series of rounds. There is a special variable t of type integer to denote a round.

5.4.3 Repositories, Production and Consumption

Some amount of materials can be owned by each participant.

The amount of material m owned by participant u at time t is denoted as R_t^{mu} . It is defined that u is satisfied with m at time t if $R_t^{mu} \geq 1.0$. Each material can have its own production and consumption rates, denoted as pr_m and cr_m , respectively. But for the purpose of this dissertation, these are one

set of constants as shown in Table 5.6, unless it is a simulation concerning NEO where materials are divided into matters, labors and information.

Table 5.6: Common parameters for the simulations

<i>Maximum debt</i>	10.0
<i>Maximum active trades per round</i>	3
<i>Production rate pr_m for all $m \in M$</i>	3.0
<i>Consumption rate cr_m for all $m \in M$</i>	0.1
<i>Probability to search the 2nd hop for a partner</i>	0.2

The amount R_t^{mu} may vary during a round because of trades, and is finalized at the end of the round. The initial amount R_{t+1}^{mu} for time $t + 1$ is calculated as follows:

$$R_{t+1}^{mu} = R_t^{mu} \times (1.0 - cr_m) + Z$$

where $Z = pr_m$ if $f(u) = m$, and $Z = 0$ otherwise.

This models production and consumption of material goods such as foods, but by adjusting pr_m and cr_m , it can also model labors ($cr_m = 1.0$) or information such as a data file (very large pr_m) as commodities, which are more notable subjects of exchanges in P2P systems. Some preliminary simulations showed such material types affect the results only in a proportional way to the production and consumption rates. Therefore the author uses the constant values shown in Table 5.6 to model both real-life and P2P exchange environments.

In essence,

- By setting the consumption rate to 1.0 (100%), the simulation can deal with labor hours or other temporary resources.
- By setting the production rate infinite (or sufficiently large), the simulation can deal with resources with extremely small production cost, such as information.

5.4.4 Currencies

Currencies are valued by the amount of materials. To purchase 1.0 amount of a material, 1.0 amount of a currency is required.

Each participant u has an account in an MCS, whose balance at time t is denoted as B_t^u .

Every participant u can use WAT/ i -WAT currency systems. A_t^u is the set of tickets u has acquired by time t , and D_t^u is the set of tickets u has drawn by time t . Every ticket $k \in A_t^u$ or $k \in D_t^u$ is considered to be a sequence of endorsers, where the drawer and lender are denoted as k_0 and k_1 , respectively. Thus for every $k \in D_t^u$, it holds that $k_0 = u$.

5.4.5 Welfare

Welfare W_t^u is a value representing how well the participant u has spent their lives in the world up to time t .

$$W_t^u = \sum_{i=1}^t \sum_{m \in M} \min(R_i^{mu}, 1.0)$$

The goals of the currency systems are twofold: (1) to maximize the welfare of participants, and (2) to minimize the variability in the distribution of welfare in the world.

5.4.6 Balance

Balance B_t^u is a value representing the purchasing power of the participant u at time t . Each participant can purchase materials as long as their debt (negative component of the balance) does not exceed the predefined maximum value.

$$B_t^u = \sum_{a \in A_t^u} \text{value}(a, t) - \sum_{d \in D_t^u} \text{value}(d, t) + B_t^u$$

For the purpose of this dissertation, uses of MCS accounts and WAT/ i -WAT tickets are mutually exclusive.

5.4.7 Trades

Participants purchase 1.0 of their partners' manufactured material in one trade.

An *active trade* is to purchase a material by finding a partner. The precondition for a successful trade for a participant u and their partner u' is $R_t^{mu'} \geq 1.0 \wedge R_t^{mu} < 1.0$ where $f(u') = m$. The maximum number of active trades that is allowed to a participant in one round is predefined. In a round, all participants try to participate in active trades within such a limit.

When drawing *multiplication* tickets, participants always issue multiple number of tickets at once, each of them has a division the intended value, such that their maximum values do not exceed 1.0. Otherwise, the tickets will be unable to use.

A *passive trade* is to vend a material by the request of a partner. There is no limit to the number of passive trades per round.

5.4.8 Bankruptcy

At the end of a round, a participant may go bankrupt if their debt is equal to or greater than a specified limit, by a predefined probability p . This

probability is called *bankruptcy rate* henceforth in this dissertation, but the actual probability p' for any participant to go bankrupt during a simulation is dependent on the probability p'' for their debt to reach the limit.

$$p' = 1 - \prod_{i=1}^t (1 - p \times p'')$$

The procedure for a bankruptcy of a participant u at time t is as follows:

- Remove all (u, x) and (x, u) from the acquaintance relation K .
- Add (u, x) and (x, u) to K for a random partner x .
- Set $B_t^u = 0$ (the debt becomes inaccessible).
- Empty D_u after for all $k \in D_u$ and $x = k_1$, adding k to D_x and removing u from k (the security rule).
- Empty A_u after for all $k \in A_u$ and $x = k_0$, removing k from D_x (treated as redemption).

In other words, u resets their relationships with others, and starts again.

Chapter 6

Results

6.1 Simulated Results

6.1.1 Mass-Market MCS

First, the author simulates a mass-market MCS, where participants can find trade partners from the whole population. This simulation is important, because such an MCS seems to be where electronization of most complementary currencies, as well as most P2P barter currencies, are heading.

This simulation will show some deficits of the design.

Mass-Market Partnership

Participants choose their partners randomly from the whole population. The acquaintance relation K is not altered after trades.

Welfare Distributions and Bankruptcy Rates

Figure 6.1 shows box-and-whisker plots of the welfare distributions after 500 rounds in the simulated mass-market MCS with different bankruptcy rates.

A box-and-whisker plot shows the median as a thick line, and a box is drawn around them to cover first and third quartiles. Small circles represent values that are extreme.

The plots show that although the upper extreme gets lowered as the bankruptcy rate grows, the median welfare increases and the variability decreases, suggesting as if the currency gets closer to achieving its goals. This paradoxical result is explained by a consequence of bankruptcies (and re-joining the system); the participants regain the purchasing power that they were deprived of. This new supply of currency while the population remains constant indicates that they share higher level of debt than before, but the fact is not obvious to them because their welfare increases.

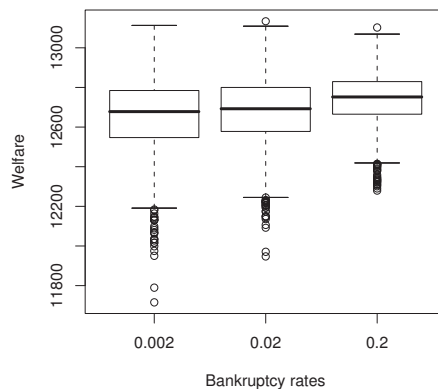


Figure 6.1: Welfare distributions in mass-market MCS

Welfare Distributions with Whitewashers

Whitewashers are represented as a group in the population which has higher bankruptcy rate (0.2) than that of regular users (0.002).

Figure 6.2 shows box-and-whisker plots of the welfare distributions in the simulated mass-market MCS with different ratios of whitewashers to the whole population. The plots show that whitewashers are always at better welfare levels, and having more whitewashers has slightly better effects on the welfare of regular users, too; this indicates that there is no pressure within the system to stop the increasing number of whitewashers. The author believes that this is an indication of possible moral hazards.

6.1.2 Small-World MCS

A mass-market MCS is costly to operate in real life because participants need to be provided information on the whole population to search for their prospective partners. Once a partner is found, there is also a difficulty in communicating with someone they do not know. Partners should be more easily found among one's acquaintances.

Small-World Partnership

In *small-world partnership*, participant u chooses their partner randomly from $\{x | (u, x) \in K\}$ in most cases, or from $\{x | (u, z) \in K \wedge (z, x) \in K\}$ by a predefined probability. In the latter case, (u, x) and (x, u) are added to K if the trade was successful and they are not already in K ; the number of prospective partners increases as they continuously participate in trades.

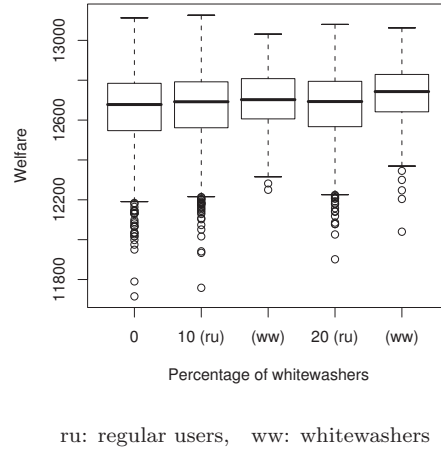


Figure 6.2: Welfare distributions in mass-market MCS with whitewashers

Figure 6.3 shows how the different sizes of population affect the level of welfare in a mass-market MCS, as a reference to what we should expect as consequences of the growing number of prospective partners in a small-world MCS. The figure shows that the welfare increases logarithmically to the population, which is attributed to the increasing chances of finding available partners. The change in the difference between mean and median welfare indicates that there is less variability in the distribution of welfare as the population grows.

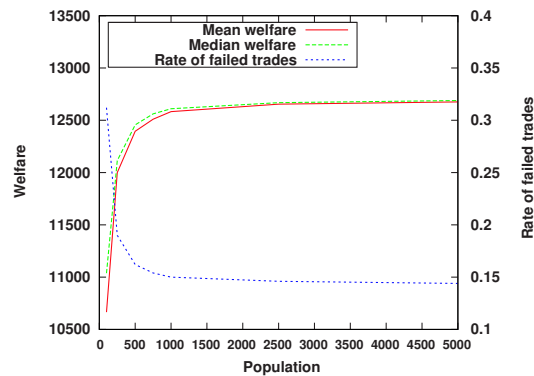


Figure 6.3: Population, welfare and rate of failures

These correlations between population and welfare can be used to punish whitewashers in a small-world MCS; they need to start again with a small

set of prospective partners every time they go bankrupt, and their levels of welfare are predicted to be less than those of regular users.

Welfare Distributions and Bankruptcy Rates

Figure 6.4 shows box-and-whisker plots of the welfare distributions in the simulated small-world MCS with different bankruptcy rates. The plots show

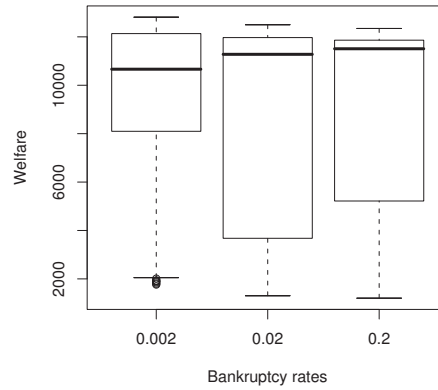


Figure 6.4: Welfare distributions in small-world MCS

that although there are more participants with lower levels of welfare as the bankruptcy rate grows, the median welfare still increases.

Welfare Distributions with Whitewashers

Figure 6.5 shows box-and-whisker plots of the welfare distributions in the simulated small-world MCS with different ratios of whitewashers to the whole population. The plots indicate that the punishment is in working for whitewashers, but the regular users may still be indifferent to the increasing number of bad users, if not welcoming them.

6.1.3 The WAT System

Preconditions

Participants choose their partners by small-world partnership, and all evasive actions with respect to *regular* tickets (EV1, EV2 and EV3) are implemented.

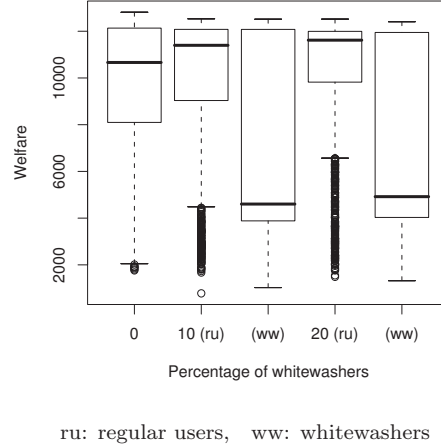


Figure 6.5: Welfare distributions in small-world MCS with whitewashers

Welfare Distributions and Bankruptcy Rates

Figure 6.6 shows box-and-whisker plots of the welfare distributions in the simulated WAT System with different bankruptcy rates. The plots show that the distribution of welfare drastically changes as the bankruptcy rate grows, which is attributed to the security rule where everyone has a chance of taking over the past partners' debt. While participants are not allowed to be indifferent to the growing number of bankruptcies, it looks as if the system is unstable as to the presence of such misbehaviors.

Welfare Distributions with Whitewashers

Figure 6.7 shows box-and-whisker plots of the welfare distributions in the simulated WAT System with different ratios of whitewashers to the whole population. The plots indicate that regular users are slightly affected by the growing number of whitewashers.

6.1.4 Comparative Study on i -WAT (regular tickets)

Preconditions

A participant u chooses their partner u' by small-world partnership. If the trade was successful with a ticket k and $(k_0, u') \notin K$, then (k_0, u') and (u', k_0) are added to K . This is to be compatible with the semantics of the i -WAT trust model, in which the drawers of tickets are always responsible for validating trades, and every receiver of a ticket becomes an acquaintance of the drawer.

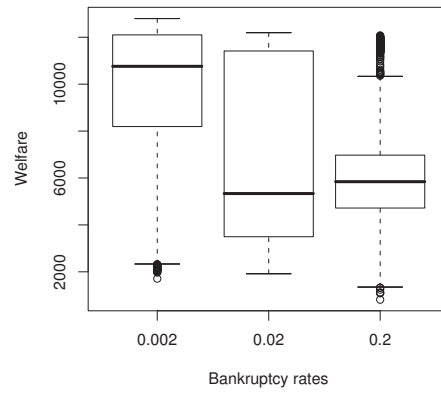
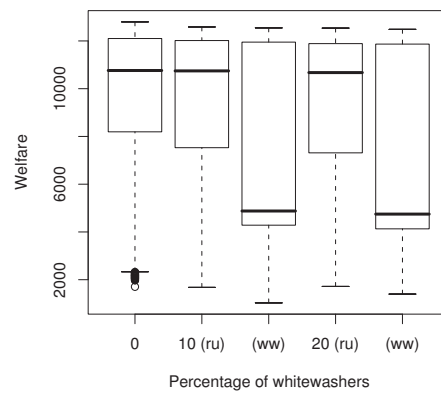


Figure 6.6: Welfare distributions in the WAT System



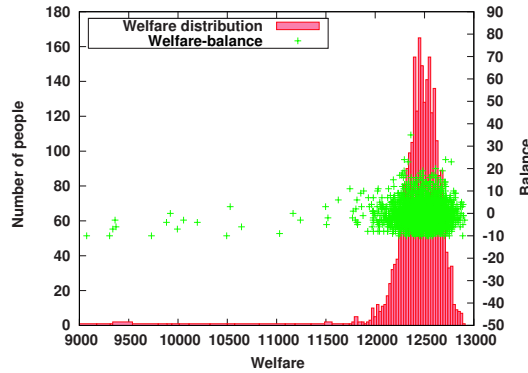
ru: regular users, ww: whitewashers

Figure 6.7: Welfare distributions in the WAT System with whitewashers

Every simulation is compared with *placebo* cases where instead of adding the drawer k_0 , a randomly chosen participant is added to the set of the receiver's acquaintances; this makes the effects of the *i*-WAT trust model, if any, independently measurable from the effects of growing population.

Base Results

Figure 6.8 shows the welfare distribution and the scatter plot of welfare-balance in the simulated *i*-WAT with the bankruptcy rate of 0.002. Mean and median welfare, SIQR (Semi Inter-Quartile Range) and the number of bankruptcies are also listed, comparing these figures with those of a placebo case. The scatter plot shows that purchasing power is not affected by the

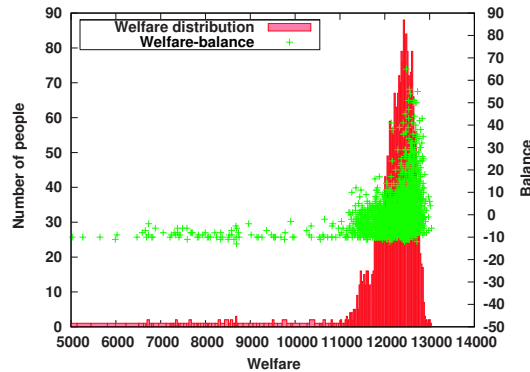


	<i>i</i> -WAT users	Placebo users
Mean welfare	12461	12403
Median welfare	12487	12537
SIQR	113.5	175.6
Bankruptcies	28	62

Figure 6.8: Welfare distribution in *i*-WAT (bankruptcy rate: 0.002)

level of welfare for majority of participants. Both SIQR and the number of bankruptcies are remarkably small for real *i*-WAT than the placebo case, suggesting that the *i*-WAT trust model has some kind of effects over the stability of the system.

Figure 6.9 shows the plots for the same settings, but without evasive actions. They select the tickets to use on a first-in, first-out basis. The scatter plot shows a slight positive correlation between welfare and purchasing power. Evasive actions seem to have an effect of reducing the level of such a correlation. It is also suggested that effects of the *i*-WAT trust model over stability of the system have something to do with evasive actions of the participants.



	<i>i-WAT users</i>	<i>Placebo users</i>
<i>Mean welfare</i>	12153	12175
<i>Median welfare</i>	12331	12393
<i>SIQR</i>	248.5	286.3
<i>Bankruptcies</i>	124	125

Figure 6.9: Welfare distribution in *i-WAT* without evasive actions (1)

High Bankruptcy Rate

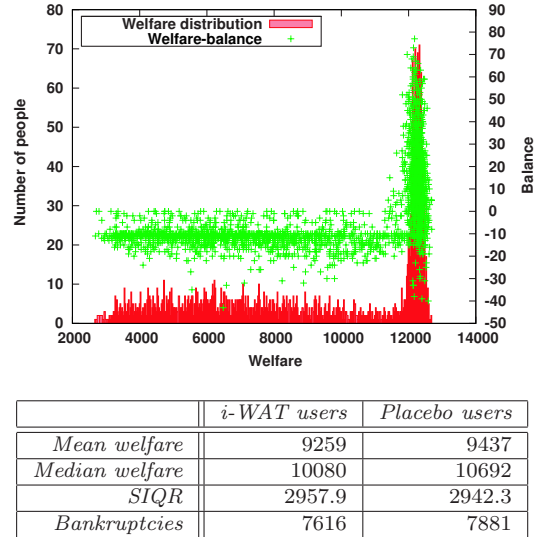
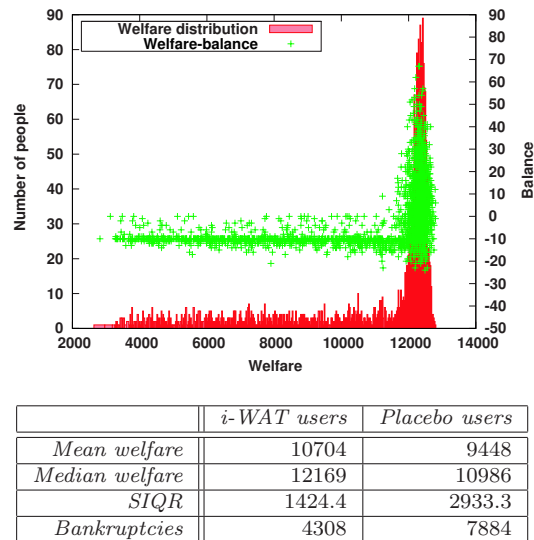
Figure 6.10 shows what happens if the bankruptcy rate is raised to 0.02 while the evasive actions remain uninstalled. We see a large number of bankruptcies and large variability both in welfare and the purchasing power. Note that the balance of some participants are below -10, the set limit for one's debt. This is an evidence that there were indeed bankruptcies, and the security rule of the system has been applied; some participants had to take over the debt of others who went bankrupt.

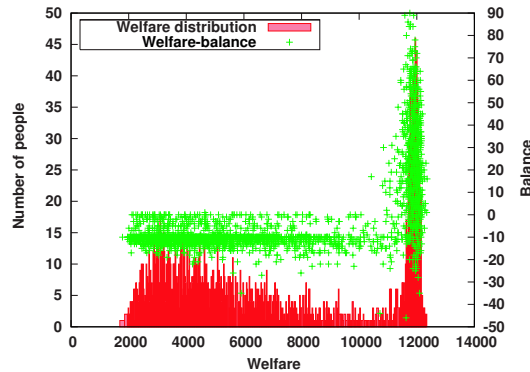
EV1 (elimination)

Figure 6.11 shows the effects of EV1 (elimination), which is added to the previous settings. Comparison with the placebo case indicates that this action is particularly effective with the *i-WAT* trust model, reducing the number of bankruptcies almost down to half. This is because the trust model increases the chance of choosing a partner who happens to be the drawer of acquired tickets as those are always included in one's set of acquaintances.

EV1 + EV2 (stretch)

Figure 6.12 shows the effects of EV2 (stretch), which is added to the previous settings. The plots suggest that this action is only a local optimization (because the participants may be exposed to more risks without it), and it has an overall negative effects on the levels of welfare, its distribution and

Figure 6.10: Welfare distribution in *i*-WAT without evasive actions (2)Figure 6.11: Welfare distribution in *i*-WAT with EV1



	<i>i-WAT users</i>	<i>Placebo users</i>
<i>Mean welfare</i>	7142	9462
<i>Median welfare</i>	6119	10723
<i>SIQR</i>	3900.0	2932.6
<i>Bankruptcies</i>	11046	7692

Figure 6.12: Welfare distribution in *i-WAT* with EV1 and EV2

the number of bankruptcies. It is even more so for real *i-WAT* than the placebo case.

To verify the hypothesis that EV2 is a local optimization, let us investigate the case where two groups are mixed in one simulation: one whose members always try to use the tickets with the longest chain of endorsements, and one whose members randomly select the tickets to use. Figure 6.13 shows the welfare distributions of the two groups when they are mixed in the ratio of 50:50. The plots indicate the presence of some risks of not applying EV2; it is effective to the presence of those who do not apply this action, and if someone is applying this action, other participants will want to apply it too, to avoid being disadvantageous. This has a collective effect of introducing a tendency to the system that redemption is deferred, which makes the participants stay longer in the state of being in heavy debt, increasing the chance of bankruptcies.

This point is further explained with Figure 6.14, which shows the distributions of the chain lengths of those tickets found redeemed in simulations, with different levels of evasive actions. The intended outcome of EV2 is stretched chains of endorsements, and there indeed are a small number of tickets with extremely long chains, but the chains of a huge number of them remain short. While trying to stretch the chains, redemption is deferred, increasing the chance that the drawer of a ticket goes bankrupt. When they do, the chains of those tickets which have been acquired by them are cut short, because these tickets are treated as if they are redeemed. Note that the participants who try to stretch the chains are not necessarily the ones who need to go bankrupt; such a consequence is taken care of by others,

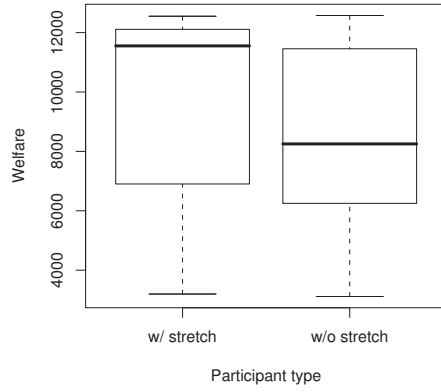
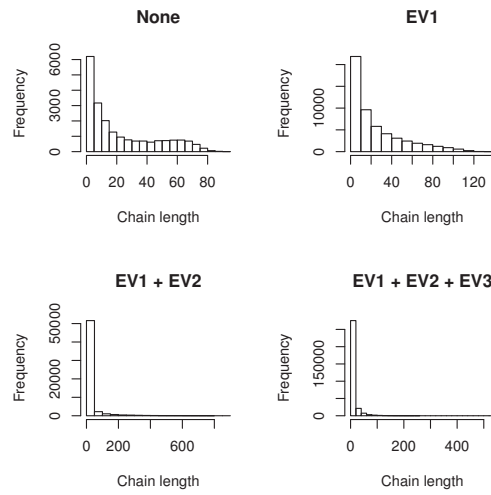


Figure 6.13: Competing welfare distributions with or without EV2



Number of redeemed tickets			
<i>None</i>	<i>EV1</i>	<i>+EV2</i>	<i>+EV3</i>
20771	53405	57125	311367

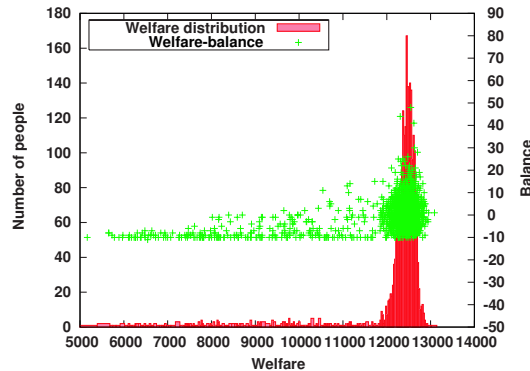
Figure 6.14: Distributions of chain lengths with evasive actions

and stretching does have an effect of reducing the risk of taking over the debt of others. Once some participants start taking EV2, the occurrence of bankruptcies increases, increasing the risk that participants need to take over someone's debt. This would result in more participants being interested in taking EV2 to avoid shorter chains.

This is a form of a Tragedy of the Commons[34], but it is an acceptable protective behavior because its negative effects can be absorbed by increasing the chances of eliminating tickets.

EV1 + EV2 + EV3 (matchmaking)

Figure 6.15 shows the effects of EV3 (matchmaking). Adding this action



	<i>i-WAT users</i>	<i>Placebo users</i>
<i>Mean welfare</i>	12105	11536
<i>Median welfare</i>	12456	12470
<i>SIQR</i>	139.6	238.8
<i>Bankruptcies</i>	916	2390

Figure 6.15: Welfare distribution in *i-WAT* with EV1, EV2 and EV3

has a positive impact overall, especially with the *i-WAT* trust model. This is because the model ensures that choosing a partner among the drawers of one's acquired tickets is compatible with the small-world partnership.

Figure 6.16 shows the frequencies of trade types with or without evasive actions. It indicates that more tickets are eliminated during a simulation if participants take evasive actions.

Tolerance of Whitewashers

Figure 6.17 and Figure 6.18 show box-and-whisker plots of the welfare distributions in the simulated *i-WAT* with different bankruptcy rates and with different ratios of whitewashers to the whole population, respectively. *i-WAT* shows very small variability for the majority of regular users, and they are slightly affected when the number of whitewashers grows.

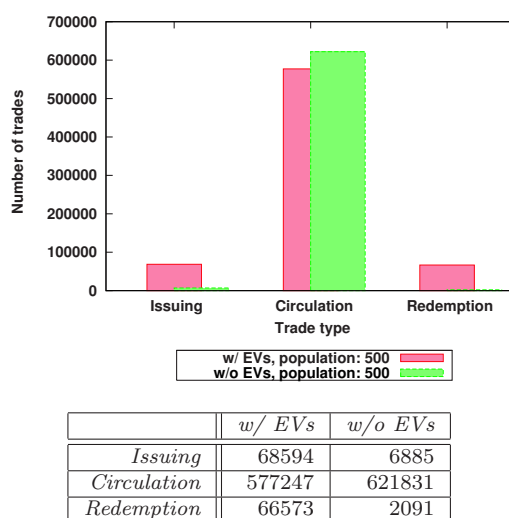


Figure 6.16: Frequencies of trade types w/ or w/o evasive actions

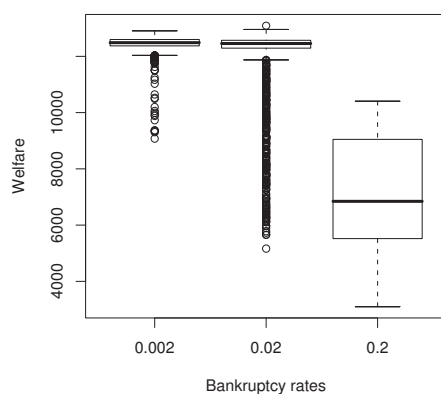


Figure 6.17: Welfare distributions in *i*-WAT

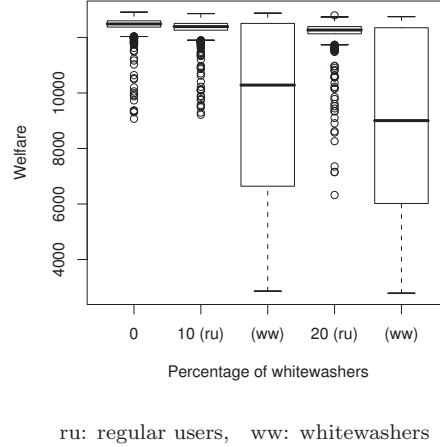


Figure 6.18: Welfare distributions in i -WAT with whitewashers (1)

Figure 6.19 shows box-and-whisker plots of the welfare distributions in the simulated i -WAT with and without evasive actions, and with different ratios of whitewashers to the whole population. Those plots show regular users only. The plots indicate that those evasive actions out of self-interest can contribute to decreasing the number of bankruptcies and raising the overall welfare of the community.

Figure 6.20 shows how the total debt of the world grows differently by the simulation settings we have investigated.

6.1.5 Comparative Study on i -WAT (variance over time)

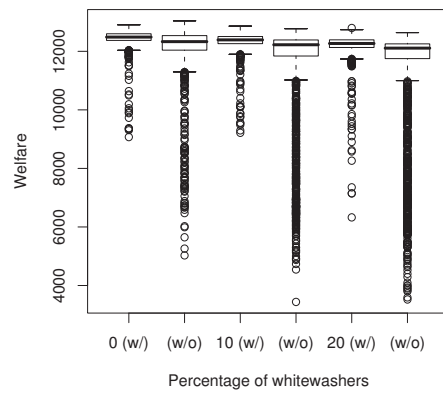
Base Results for Smaller Population

Due to insufficiency of the computing resources, a smaller world is used for a series of simulations involving *reduction* or *multiplication* tickets.

Figure 6.21 shows the welfare distribution of i -WAT users in the world population of 500, where the bankruptcy rate is set to 0.002. Figure 6.22 shows the box-and-whisker plots of welfare distributions with different levels of the presence of whitewashers in the same settings. These figures are used as references when investigating the outcome of the simulations involving variance tickets.

Over-Time Rates

For simplicity, we divide the population into some groups: each for those who can issue *regular* tickets only, *reduction* tickets only and *multiplication*



w/: with evasive actions, w/o: without evasive actions

Figure 6.19: Welfare distributions in *i*-WAT with whitewashers (2)

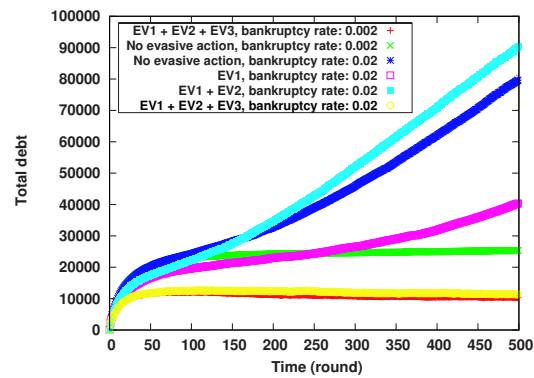
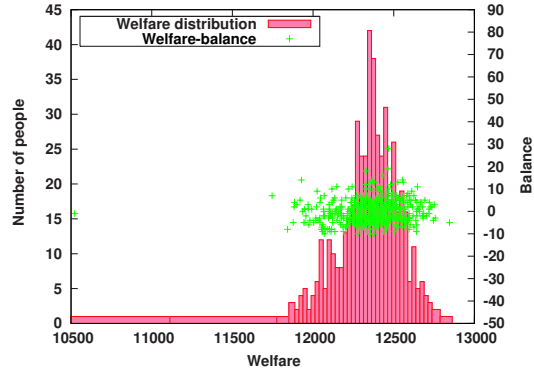
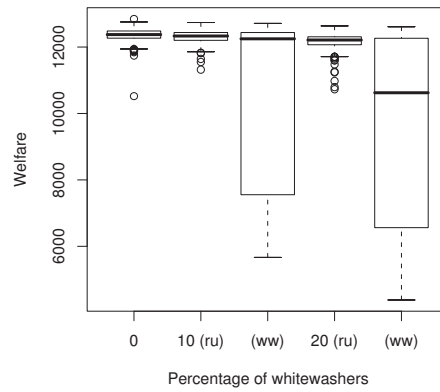


Figure 6.20: Total debt with or without evasive actions



	<i>i-WAT users</i>
<i>Mean welfare</i>	12360
<i>Median welfare</i>	12375
<i>SIQR</i>	107.9
<i>Bankruptcies</i>	3

Figure 6.21: Welfare distribution in *i-WAT* (population: 500)



ru: regular users, ww: whitewashers

Figure 6.22: Welfare distributions with whitewashers (population: 500)

tickets only. They can receive and use any kinds of tickets. If necessary, the groups may be further divided by the set of actions they take against risks or the variance ratio of the tickets they use.

Figure 6.23 shows how different over-time rates affects the mean welfare of the groups. The x axis of the graph is the absolute value of the rates; if the rate is 0.1, it means that the values of *multiplication* tickets increase by 10% per round, and those of *reduction* tickets decrease by 10%. Approximately $\frac{1}{3}$ each of the population belong to each of the groups. We simulate their behaviors when they take no further evasive actions than EV1, EV2 and EV3 (*not optimized*), and when they take EV4 and GR1 as well (*optimized*). The

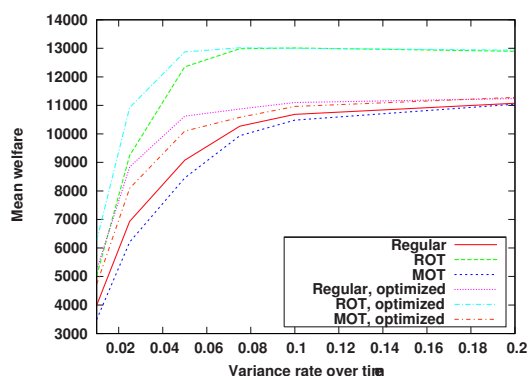


Figure 6.23: Over-time rates and mean welfare

figure shows that the mean welfare of all groups, whether their behaviors are optimized or not, grow in logarithmic curves as the over-time rate increases. The group of *reduction*-ticket issuers shows especially high mean welfare, followed by *regular*-ticket issuers, and then *multiplication*-ticket issuers. The reason is explained as follows: *reduction*-ticket issuers tend to be able to issue more tickets as their debts decrease over time, whereas the inverse is true for *multiplication*-ticket issuers.

On the other hand, optimization is particularly effective for *regular* and *multiplication*-ticket issuers. It is not effective for *reduction*-ticket issuers if the over-time rate is over 0.08 in the simulation settings.

Figure 6.24 shows how the total debt of the world is affected by the different over-time rates and optimization. The total debt approaches that of the reference settings as the over-time rate increases. The effect of optimization to the amount of debt is not apparent, but it tends to influence to lower the debt after 200 rounds.

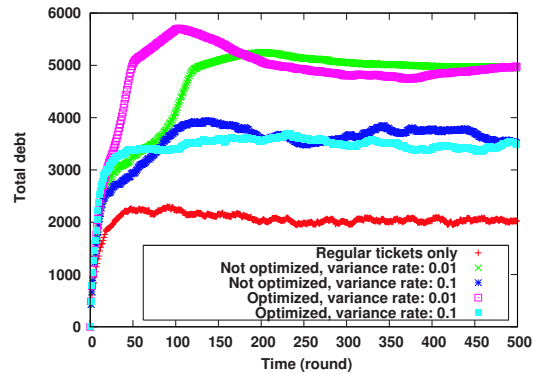


Figure 6.24: Total debt with different over-time rates

Reduction/Multiplication Ratios

Figure 6.25 shows how different reduction ratios affect the mean welfare of three groups: ratio 1.0 (*regular-ticket*) issuers which makes up 50% of the population, ratio 0.5 (*reduction-ticket*) issuers which makes up 25% of the population, and ratio 0.0 (*vanishing reduction-ticket*) issuers which makes up the rest of the population. Semi-optimized means that EV4 only is added to their behaviors (GR1 is excluded). The mean welfare of the reference settings is shown by the dotted line. The figure indicates that the mean

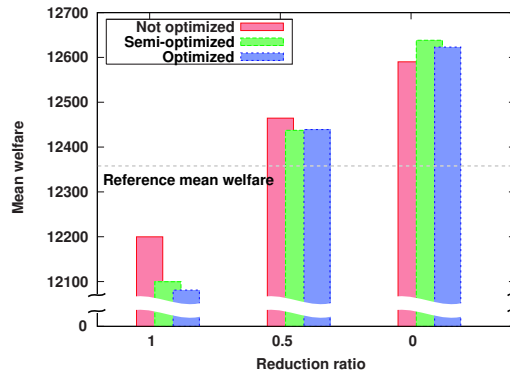


Figure 6.25: Reduction ratios and mean welfare

welfare increases as the values are more reduced. The mean welfare of the ratio 0.5 and 0.0 issuers are well over the reference mean welfare.

Figure 6.26 shows how different multiplication ratios affect the mean welfare of three groups: 50% ratio 1.0 (*regular-ticket*) issuers, 25% ratio 1.5 (*multiplication-ticket*) issuers, and 25% ratio 2.0 (*doubling multiplication-ticket*) issuers. The mean welfare of the reference settings is shown by the

dotted line. The mean welfare of all groups are well below the reference mean

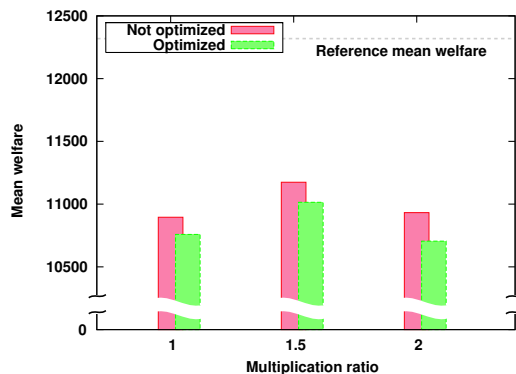


Figure 6.26: Multiplication ratios and mean welfare

welfare. This is attributed to the fact that *multiplication* tickets have effects of suppressing trades; increased debt of the *multiplication*-ticket issuers allow them to issue smaller number of tickets compared to issuers of other ticket types. Optimization further accelerates this tendency because usage of such tickets at all is postponed if possible until the effective values of the tickets reach their maximum, by the self-interest of those who have received them.

Effects of Optimization

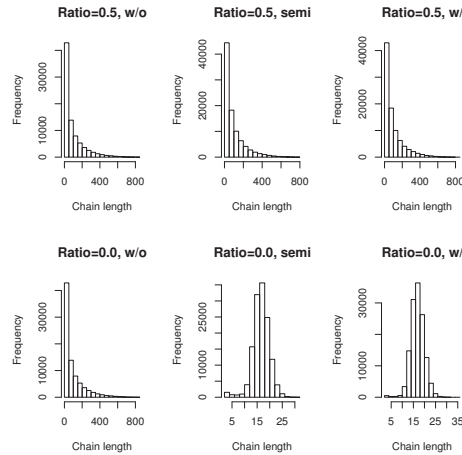
Figure 6.27 shows the distributions of the chains lengths for the tickets used by ratio 0.5 and 0.0 issuers in the previous simulation involving *reduction* tickets.

EV4 (forwarding) suggests that there are preferences to use tickets with higher reduction ratio (ratio 0.0 in this case) to minimize the losses caused by holding these tickets. Ratio 0.0 tickets tend to be forwarded to others with the presence of EV4, and very large number of them are forwarded to 15~20 participants, enabling them to participate in active trades, and contributing to raising their welfare.

GR1 has an effect of lengthening the chain, because holders of *reduction* tickets will wait until their values reach the specified minimum before trying to use them against the drawers. These tickets are less frequently redeemed, and smaller number of such tickets were in circulation comparing to the semi-optimized case.

As Figure 6.25 shows, EV4 is only positively affecting for ratio 0.0 tickets. EV4 seems to be a local optimization which has a slightly negative effect on the welfare of all.

As an evidence, Figure 6.28 shows the box-and-whisker plots of welfare distributions where two groups are involved in one simulation: 50% *regular*-



Number of redeemed tickets			
	<i>w/o</i>	<i>semi</i>	<i>w/</i>
<i>Ratio=0.5</i>	82000	92007	90666
<i>Ratio=0.0</i>	82000	133073	131941

w/o: not optimized, *semi*: semi-optimized, *w/*: optimized

Figure 6.27: Distributions of chain lengths for *reduction* tickets

ticket issuers and 50% vanishing *reduction*-ticket issuers. A half of each takes EV4. The plots show that EV4 does avoid the risk of contributing too much to reducing the debts of the issuers. But again, this action has a collective effect of deferring redemptions and suppressing trades.

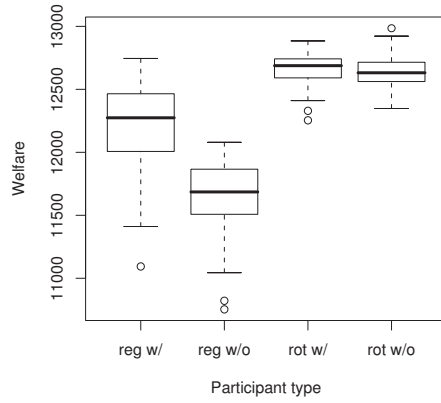
Figure 6.29 shows the distributions of the chain lengths for the tickets used by ratio 1.5 and 2.0 issuers in the previous simulation involving *multiplication* tickets. Effects are not at all clear on the distributions of the chain lengths, other than that the total number of redeemed tickets is reduced by the optimization; for *multiplication* tickets, both EV4 and GR1 have effects on postponing the usage of them, thus increasing the debt of the issuers over time, which then disallows them to issue more tickets.

Figure 6.30 shows the box-and-whisker plots of welfare distributions where two groups are involved in one simulation: 50% *regular*-ticket issuers and 50% doubling *multiplication*-ticket issuers. Behaviors of a half of each are optimized. The plots shows that the optimization has a negative effect on *regular*-ticket issuers.

Tolerance of Whitewashers

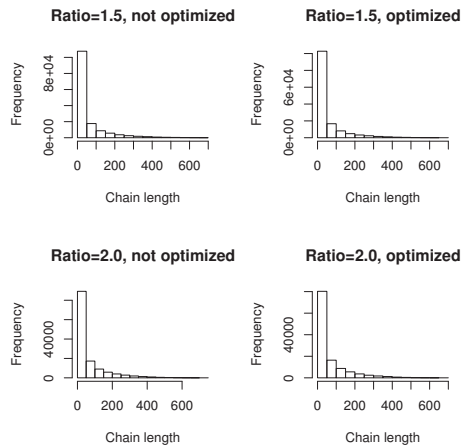
Figure 6.31 and Figure 6.34 show box-and-whisker plots for welfare distributions in which there are 10% and 20% *reduction*-ticket issuers, respectively, and all of them intend to whitewash.

As Figure 6.21 shows, the number of bankruptcies in the reference set-



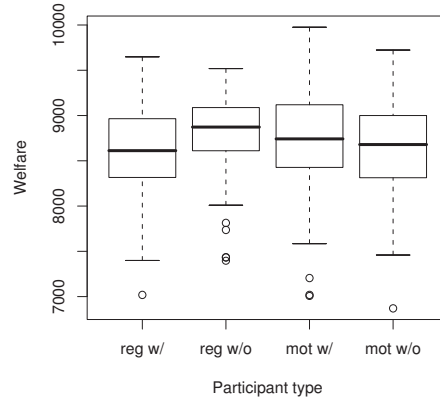
reg: *regular*-ticket issuers, rot: *reduction*-ticket issuers
w/: with EV4, w/o: without EV4

Figure 6.28: Competing welfare distributions with *reduction* tickets



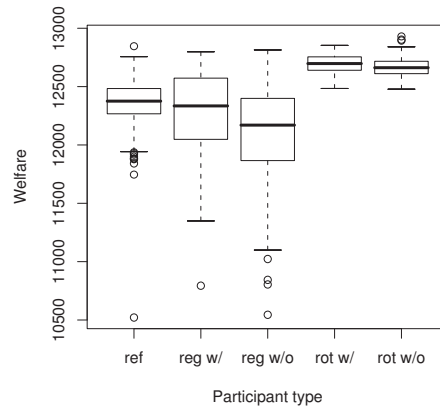
Number of redeemed tickets		
	<i>Not optimized</i>	<i>optimized</i>
<i>Ratio=1.5</i>	150980	141938
<i>Ratio=2.0</i>	133411	121253

Figure 6.29: Distributions of chain lengths for multiplication tickets



reg: *regular*-ticket issuers, mot: *multiplication*-ticket issuers
w/: with optimization, w/o: without optimization

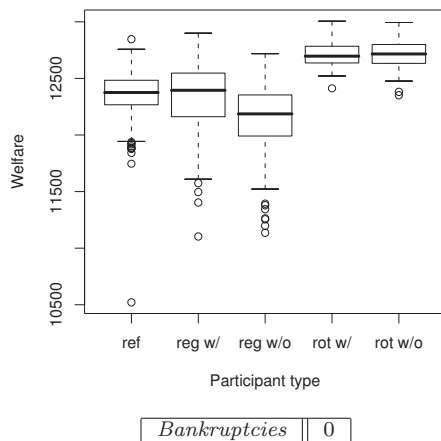
Figure 6.30: Competing welfare distributions with *multiplication* tickets



Bankruptcies | 2

ref: reference, reg: *regular*-ticket issuers, rot: *reduction*-ticket whitewashers
w/: with optimization, w/o: without optimization

Figure 6.31: Welfare distributions with 10% ROT whitewashers



ref: reference, reg: *regular*-ticket issuers, rot: *reduction*-ticket whitewashers
w/: with optimization, w/o: without optimization

Figure 6.32: Welfare distributions with 20% ROT whitewashers

tings is 3. Either of the above case shows bankruptcies of more than 3, indicating that the intended whitewashing has never happened. The reduction of the ticket values must have regulated the level of debt, and probability p'' must have been kept small. Usage of *reduction* tickets has an effect of reducing chances of bankruptcies.

Figure 6.33 and Figure 6.34 show box-and-whisker plots for welfare distributions in which there are 10% and 20% *multiplication*-ticket issuers, respectively, and all of them intend to whitewash.

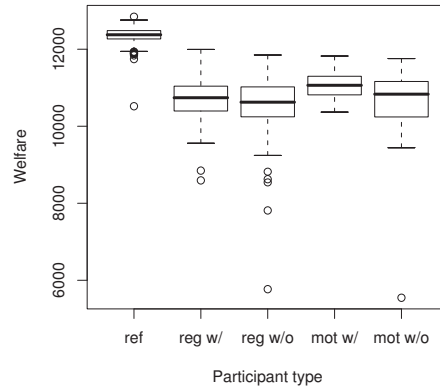
These plots indicate that whitewashing by *multiplication*-ticket issuers has a large negative impact on the welfare of all.

6.1.6 Economics in the Presence of Replicators

Overview

In this simulation about economics in the presence of replicators, the population is evenly divided into three kinds of professions: providers of physical matters such as foods (atom providers), providers of labors (presence providers) and providers of information (bit providers). Each comes with a different set of production and consumption rates, as shown in Table 6.1.

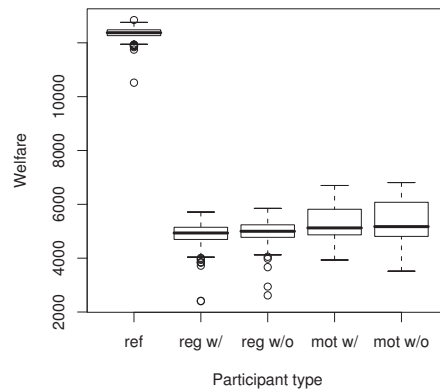
Presence providers has equal productivity as atom providers, but their products cannot be stored for future use. Bit providers show extremely high productivity, but likewise, the information has to be regenerated in each round.



<i>Bankruptcies</i>	36
---------------------	----

ref: reference, reg: *regular*-ticket issuers, mot: *multiplication*-ticket whitewashers
w/: with optimization, w/o: without optimization

Figure 6.33: Welfare distributions with 10% MOT whitewashers



<i>Bankruptcies</i>	890
---------------------	-----

ref: reference, reg: *regular*-ticket issuers, mot: *multiplication*-ticket whitewashers
w/: with optimization, w/o: without optimization

Figure 6.34: Welfare distributions with 20% MOT whitewashers

Table 6.1: Product types and production/consumption rates

	<i>Products</i>		
	<i>Atoms</i>	<i>Presences</i>	<i>Bits</i>
<i>Production rate</i>	3.0	3.0	250.0
<i>Consumption rate</i>	0.1	1.0	1.0

MCS in the Presence of Replicators

Figures 6.35 and 6.36 show how the welfare distributions differ among different professions of participants in case of a mass-market MCS. The box-and-whisker plots in Figure 6.36 are made horizontal so that it is easier to compare them with the histograms in Figure 6.35.

The plots show that those whose commodities are information experience significantly high welfare than others. On the other hand, those whose commodities are labors seem to experience significantly low welfare. The scatter plot of welfare-balance in Figure 6.35 indicates that there is a strong asymmetry in the purchasing powers of participants, which resulted from the extremely reproductive nature of information; the bit providers never run out of their commodities, and the currency tends to be accumulated in their accounts.

Figure 6.37 shows how the differences grow as the time proceeds. It shows that the differences are in the growth rates of welfare. The three professions are more separated in terms of their welfare as the time proceeds.

i-WAT in the Presence of Replicators

Figures 6.38 and 6.39 show how the welfare distributions differ among different professions of participants in case of an *i*-WAT currency.

Although differences are smaller, those whose commodities are information still tend to experience significantly high welfare than others. There is a small correlation between having a high purchasing power and being a bit provider.

Equalizing this situation would require more drastic changes in how the economy works in this world.

NEO in the Presence of Replicators

Figures 6.40 and 6.41 shows how the welfare distributions differ among different professions of participants in case of NEO, where information is not sold but shared for free, and the bit providers are appreciated by the society's allowance to them to issue *reduction* tickets.

In this simulation, a bit provider is allowed to issue ratio 0.0 (vanishing) *reduction* tickets with the over-time rate of -0.1, and a presence provider is

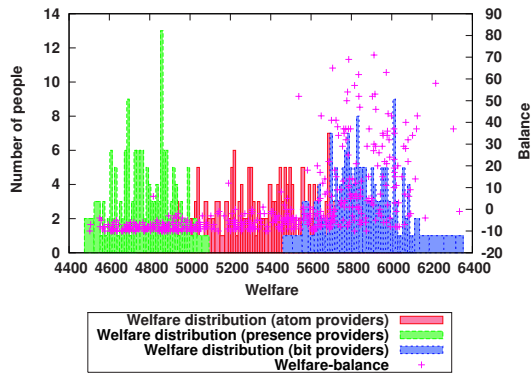


Figure 6.35: Welfare distributions for different professions (MCS) (1)

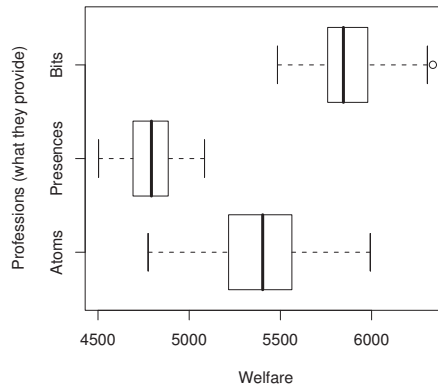


Figure 6.36: Welfare distributions for different professions (MCS) (2)

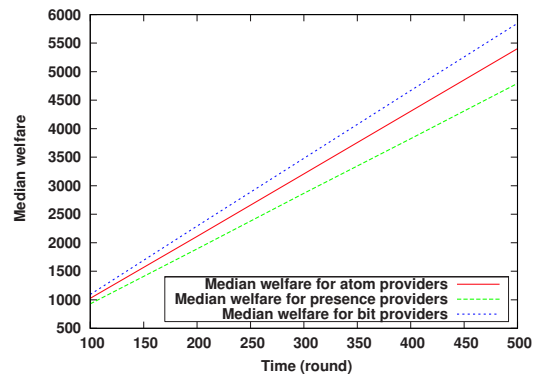


Figure 6.37: Growth of median welfare (MCS)

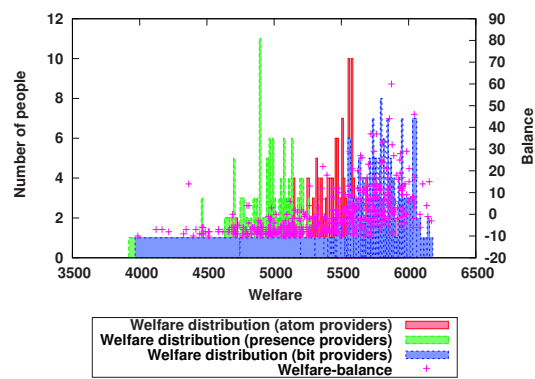


Figure 6.38: Welfare distributions for different professions (*i*-WAT) (1)

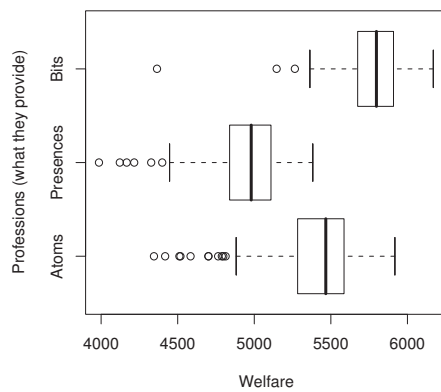


Figure 6.39: Welfare distributions for different professions (*i*-WAT) (2)

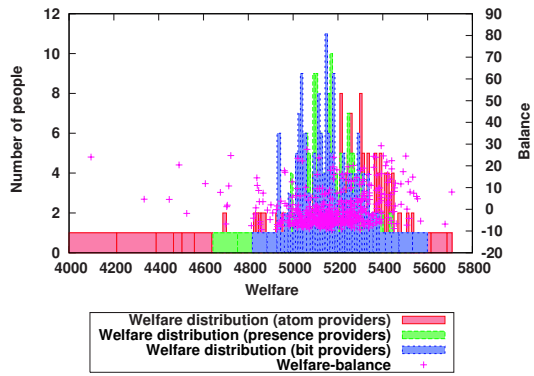


Figure 6.40: Welfare distributions for different professions (NEO) (1)

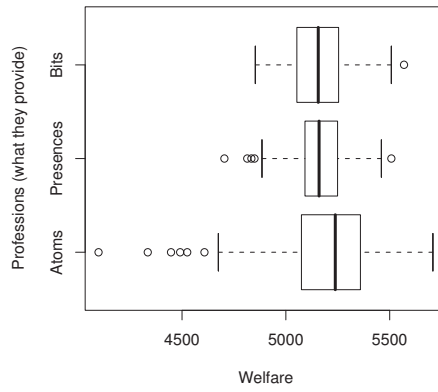


Figure 6.41: Welfare distributions for different professions (NEO) (2)

allowed to issue ratio 0.5 *reduction* tickets with the over-time rate of -0.1. The production and consumption rates for the commodity of a bit provider, which is something different from information now but perhaps a labor to design new information, are set to 1.5 and 1.0, respectively.

The plots show that the distributions of welfare overlap one another for different professions. No correlation is found between the welfare and the purchasing power of participants.

Readers may have noticed that shown welfare is significantly smaller in NEO than in other cases. This is because information is not treated as a commodity, which does not contribute in calculation of welfare, unlike other cases.

Figure 6.42 shows how the differences stay negligible in NEO as the time proceeds.

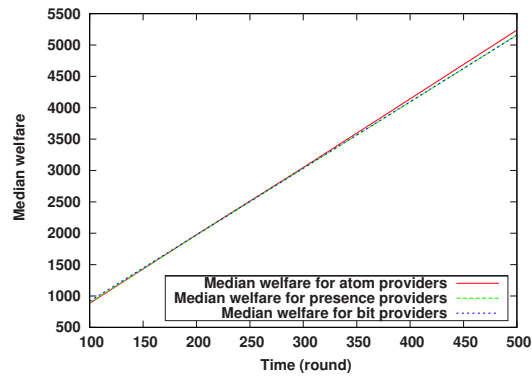


Figure 6.42: Growth of median welfare (NEO)

These results may look significant, but they have been achieved by manual adjustments of parameters for this particular simulation, and it is not at all clear how good these parameters will be in practice. We need a more autonomous, self-adapting way of allowing *reduction* tickets.

Figures 6.43 and 6.44 show results of NEO', in which peers of all professions are non-discriminatively given freedom to issue *reduction* tickets whose values are reduced down to zero.

Although this seem to produce an equally fair outcome, we think this would be difficult to achieve in practice. Taking a *reduction* ticket necessarily means a loss of some amount, and there needs to be an incentive for the receivers of these tickets to help the issuers.

We need to come up with an incentive-based, self-adapting scheme somewhere between NEO and NEO' presented in this paper.

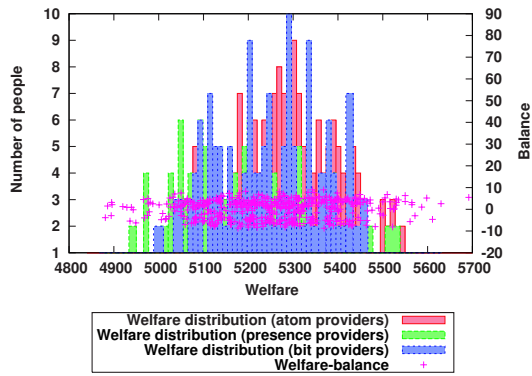


Figure 6.43: Welfare distributions for different professions (NEO') (1)

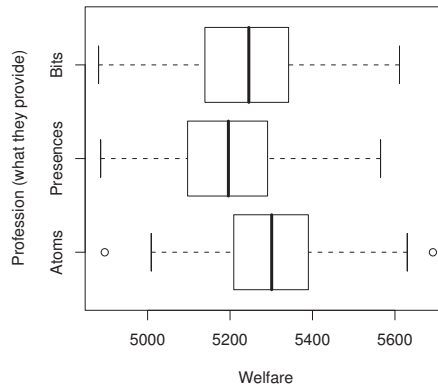


Figure 6.44: Welfare distributions for different professions (NEO') (2)

MCS⁻ in the Presence of Replicators

Readers may want to ask that, since NEO is derived as a special usage of *i*-WAT, whether it is possible or not to derive a special usage of MCS which has a similar effect.

One answer may be an MCS with a demurrage (denoted MCS⁻), which is in practice in some complementary currencies based on LETS.

Figures 6.45 and 6.46 shows how the welfare distributions differ among different professions of participants in case of a mass-market MCS with a demurrage of -1% per round for the positive balance.

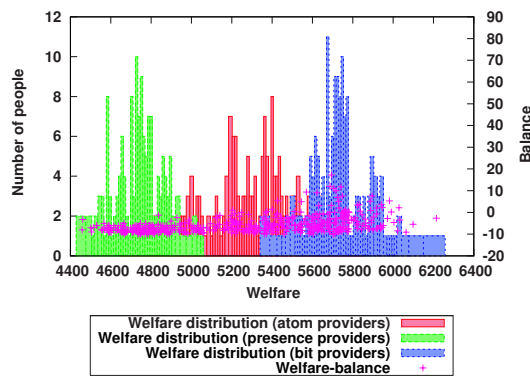


Figure 6.45: Welfare distributions for different professions (MCS⁻) (1)

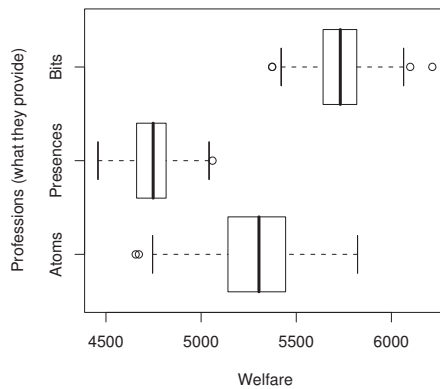


Figure 6.46: Welfare distributions for different professions (MCS⁻) (2)

Intuitively, it seems as if it would help because the purchasing power is averaged by the enforcement of a demurrage. The plots, however, show that the purchasing power is in fact averaged, but the demurrage does not seem to have such an effect of averaging welfare. This should be attributed to the

fact that the purchasing power of a bit provider is not decreased until the end of a round, and they can purchase as many commodities as they find possible during a round.

6.2 Results from Experiments

6.2.1 WIDE Hours

Table 6.2 shows some statistical information collected from an experiment in September 2003.

Table 6.2: WIDE Hours statistics from Sept. 2003

<i>Participants in some way</i>	161
<i>Total logins</i>	530
<i>Certificate</i>	406
<i>SSL + password</i>	60
<i>Plain-text password</i>	64
<i>Total trades</i>	361
<i>Revoked trades</i>	7

Table 6.3 shows the corresponding statistical information collected from an experiment in March 2004.

Table 6.3: WIDE Hours statistics from Mar. 2004

<i>Participants in some way</i>	75
<i>Total logins</i>	228
<i>Certificate</i>	206
<i>SSL + password</i>	1
<i>Plain-text password</i>	21
<i>Total trades</i>	127
<i>Revoked trades</i>	2
<i>i-WAT trades</i>	15

In the experiment conducted in September 2003, there were some pairs of participants who repetitively traded nothing with each other so that their *WIDE Power* could be boosted. This was problematic as to construction of a fair system.

In the experiment in March 2004, however, this strategy was defused by introduction of a new metric which takes variety of trade partners into consideration.

Overall, through the experiments of WIDE Hours, the author has learnt that some people are susceptible to incentives if those are presented, and some do invent strategies to boost their gains.

The *i-WAT* version of WIDE Hours is not popular among WIDE members, partly because it is burdensome to use it as it necessitate use of PGP

(see section C.1 for descriptive replies to questionnaires). Having witnessed that less techno-intimate people from Gesell Research Society Japan have been using *i*-WAT, the author believes that motivation takes more important role than technical-ease in deployment of the new technology.

6.2.2 MANA

Table 6.4 shows some statistical information collected from the experiment of MANA.

Table 6.4: MANA statistics October 26, 2003 ~ February 28, 2004

<i>Registered members</i>	123
<i>Participants*</i>	109
<i>Active participants</i>	9
<i>Total logins</i>	781
<i>Total trades</i>	253
<i>Among participants</i>	127

* Participants are those who have ever logged-in.

Unfortunately, it did not attract many people, but it worked as a proof of concept as to usage of a currency in a web-based community, as well as in linkage to the physical world with RFID.

6.2.3 Vegetable Trading

Overview

From the static analysis of the collected data, the author discovered the following:

- 24 people participated (there were 25 people, but one of them could not participate in the experiment at all because her device was not working, and replacement was deferred).
- 34 *i*-WAT tickets were issued, and 47 trades were performed (7 of them was in the waiting state for approval).
- There were two cases of redemptions (they were not reported for points).
- The maximum length of the chain of endorsement was 4 (drawer → lender → third party → the next receiver).
- The mean number of trade partners was about 3.
- The mean number of received public keys is 24 (almost all of them could receive public keys of all others; the number of public keys exceed the number of participants because of device replacements).

- The mean number of signers to their public keys was about 4.

Redemptions

Figures 6.47 and 6.48 are the examples of *i*-WAT tickets which were redeemed during the experiment. Both were drawn by participant *Eggplant 6*, and were returned directly from the lenders.

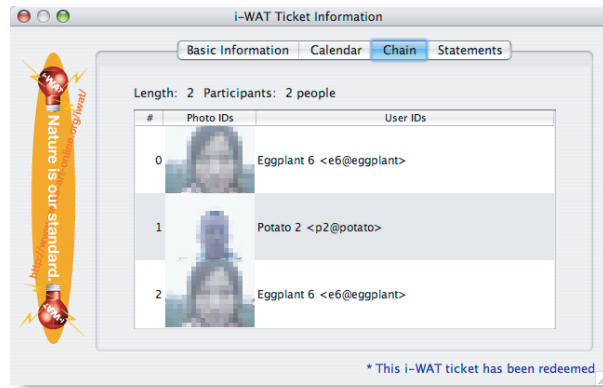


Figure 6.47: Example: redeemed ticket (1)

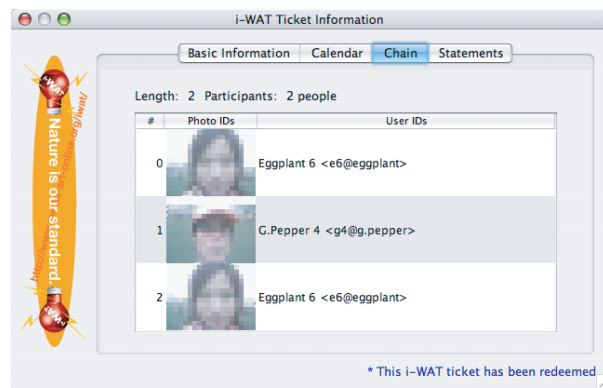


Figure 6.48: Example: redeemed ticket (2)

In the example of Figure 6.47, issuing took place at 15:07, and redemption took place at 15:28. In the example of Figure 6.48, issuing took place at 15:36, and redemption took place at 15:39. The timing of redemption was quickened as if the participants learned from their experiences.

The visual representation in this dissertation of the *i*-WAT tickets found in the experiment have been displayed by *wija+i*-WAT on a personal com-

puter instead of *wijapo+i-WAT* on the target device, using the data collected during the experiment. The same goes with the key rings.

Chains of Endorsements

Figure 6.49 shows an example of *i-WAT* ticket whose chain of endorsement involved 4 people (one of the longest chain).

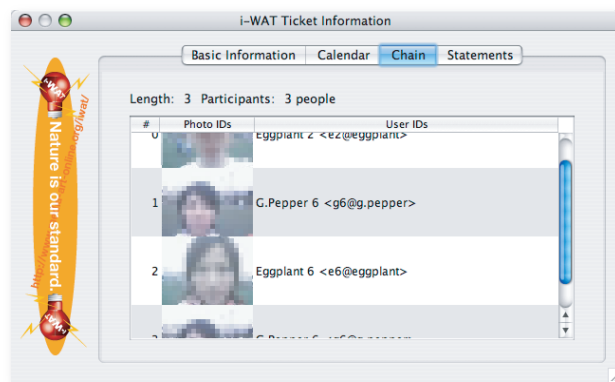


Figure 6.49: Example: one of tickets with the longest chain

As described later, the most *i-WAT* tickets never got circulated to the third parties, but a little of them did, and 3 of them, such as the one shown in Figure 6.49 were transferred to the next receivers.

Spreading Public Keys

Spreading public keys over a wireless channel was successful, and the most participants could obtain the public keys of all others without any problems.

Figure 6.50 is a visual representation of a public key ring of a typical participant. *Validity* shows the validities of the acquired public keys from the viewpoint of the participant. Green marbles indicate that the keys are considered fairly valid, and blue marbles indicate that the key belongs to the participant themselves. *Trust* shows the trustworthiness of the owners of the keys from the viewpoint of the participant. Green marbles indicate that the owners are fairly trusted, and blue marbles indicate that the owner is the participant themselves.

In general, there are more partners with whom the public keys were mutually validated than the trade partners, which makes that there are more green marbles on *Validity* column than *Trust* column.

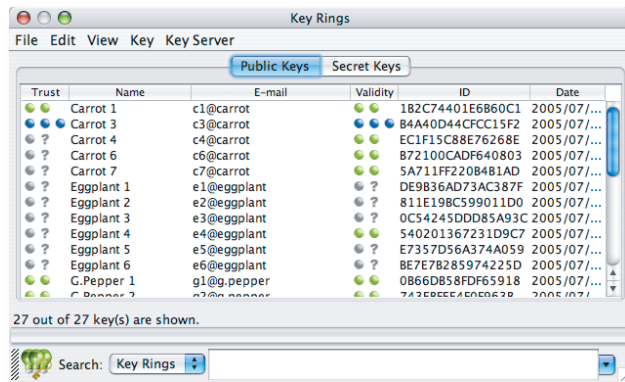


Figure 6.50: Typical public key ring

Spinning the Web of Trust

Figure 6.51 shows the resulted web of trust at the end of the experiment.

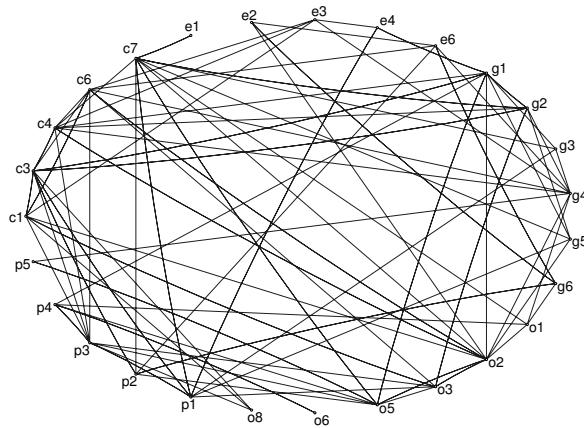


Figure 6.51: Resulted signature network (web of trust)

Figure 6.52 shows the trust network, which is effectively the barter relation generated during the experiment, because of the implementation of *mutual full trust by participation* (Property 7).

Figure 6.53 shows the validation relation created over the web of trust by the PGP trust model. This effectively defines the extent where they can readily perform *i*-WAT trades.

Table 6.5 shows some properties of the resulted web of trust and its consequences via the performed *i*-WAT trades. It is discovered from this result that implementing *mutual full trust by participation* more than doubled the

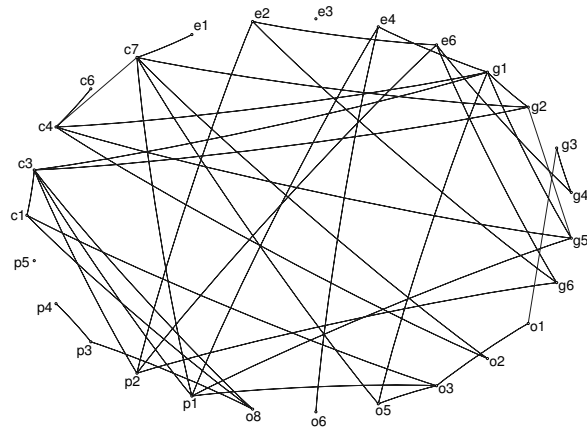


Figure 6.52: Resulted trust network (barter relation)

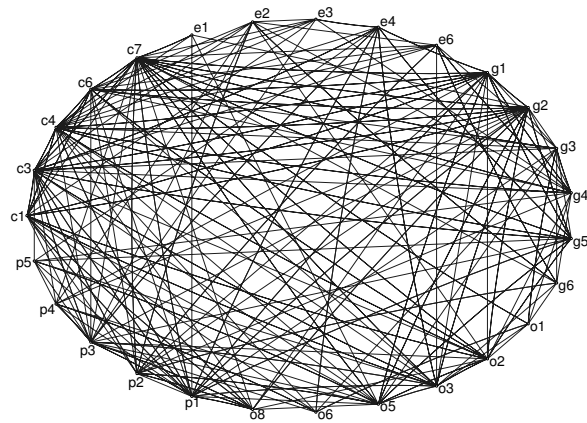


Figure 6.53: Resulted validity network (validation relation)

Table 6.5: Properties of the resulted web of trust

	<i>Signature</i>	<i>Trust</i>	<i>Validity</i>
<i>Number of vertices</i>			27
<i>Number of edges</i>	109	77	271
<i>Number of unreachable pairs</i>	0	102	0
<i>Mean distance (in hops)</i>	2.03	3.09*	1.48
<i>Maximum distance (in hops)</i>	4	6*	2

* Trust is not transitive in PGP, therefore discussing distance on the trust network is in fact meaningless.

prospective partnership.

Distribution of the Web of Trust

The validity network, in which validities of public keys are guaranteed one another, is based on physical contacts with others and trust in trade partners. This is expected to make the number of links not averaged, but it will rather show power-law distribution.

Figure 6.54 is a graph whose x axis is either the number of public keys whose owners they trust or which they think are valid, and y axis is their frequencies.

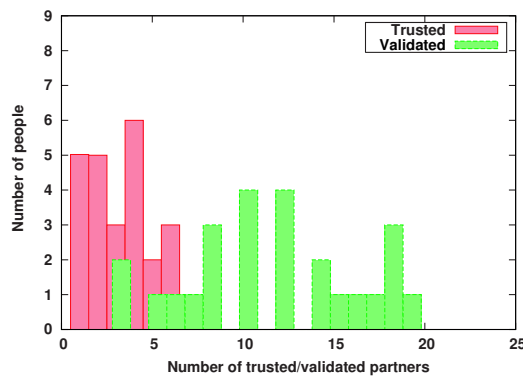


Figure 6.54: Distribution of web of trust (1)

It is understandable that validity is more averaged than trust because validity is also calculated by the PGP trust model by traversing the signature relation transitively.

Figure 6.55 is a graph whose x axis is the number of signatures participants received for their public keys, and y axis is their frequencies.

This result shows that the characteristics of power-law distribution are not visible on the signing relation among participants. This may be caused by the insufficient number of samples, as well as the artificial nature of the network of people at the experiment.

Networking

On the other hand, the trade relation using i -WAT shows more characteristics of power-law distribution.

Figure 6.56 is a graph whose x axis is the number of drawn or used tickets, and y axis is their frequencies by the number of people. Among those participants who drew or used i -WAT tickets, the ones who used just one ticket are the most common, consisting of 9 people, and those who used 4 tickets, which was the maximum, is just one.

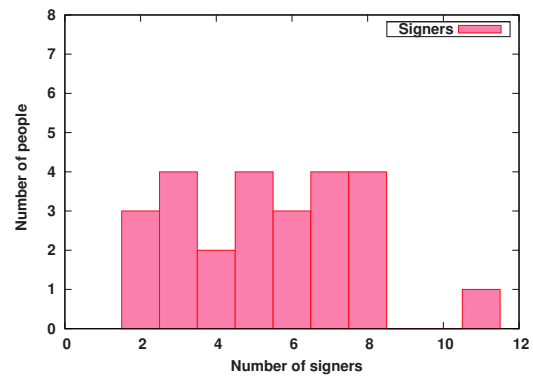


Figure 6.55: Distribution of web of trust (2)

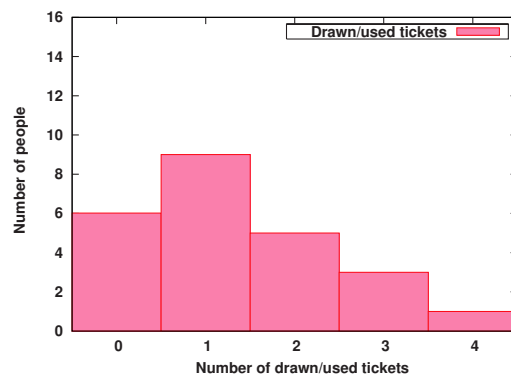


Figure 6.56: Histogram of drawn/used tickets

The same was investigated for the number of received tickets.

Figure 6.57 is a graph whose x axis is the number of received i -WAT tickets, and y axis shows their frequencies. Among those participants who received i -WAT tickets, the ones who received just one ticket are the most common, consisting of 8 people, and those who received 3 tickets, which was the maximum, are 4.

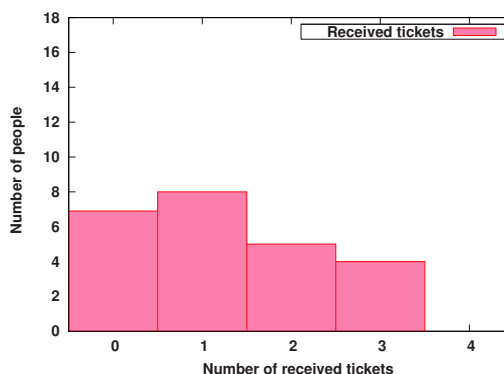


Figure 6.57: Histogram of received tickets

These results show that at least in this experiment, the trade relationship using i -WAT shows the characteristics of power-law distribution more than the web of trust by the same participants. This is perhaps trading is less easier than validating public keys because of availability of resources (vegetable in this experiment) and understanding of the i -WAT trade protocol, and therefore less influenced by being in an artificial environment.

Frequency of Trades

Figure 6.58 shows the occurrences of completed (approved) 40 trades in a time line in an accumulated way. The first few trades must have occurred during the rehearsal (there is a chance that the clocks of some devices were inaccurately set). There are only a small number of trades immediately after the game started, but as the game proceeds, the frequency increased. There are some points at which many trades were processed concurrently, including just after the ending of the game was announced. This may indicate that the procedure for i -WAT trading was steadily learned by the participants.

Distributions of Ages

Figure 6.59 is a pie chart of age-distribution among the 30 participants extracted from the replies to the questionnaire described later.

20s is the largest age group, which makes up 54%, followed by 30s (13%) and 50s (7%). 10s and 40s are represented by one person each. Ages of

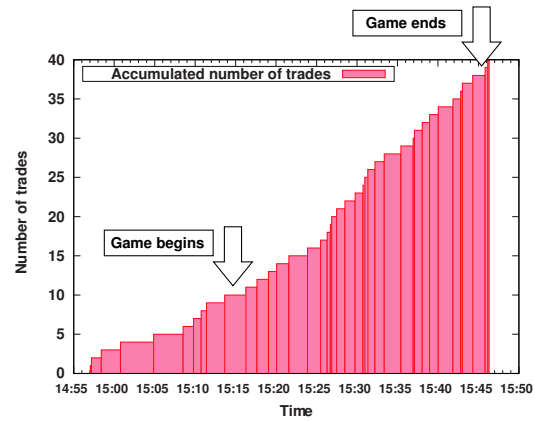


Figure 6.58: Accumulated number of trades over time

20% of the participants are unknown. Since there is in general no reason for younger generation to hide their ages, this group may consist mostly of the elder generation.

The median of the participants whose ages are known is 26.5 years old. Later analyses include cases where the participants are divided into two groups: age 26 or less (younger generation) and 27 or over (elder generation).

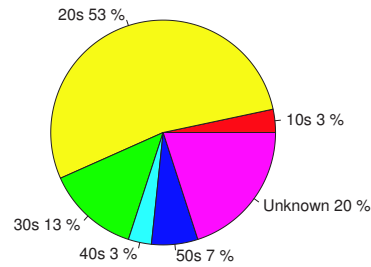


Figure 6.59: Age-distribution of participants

Genders

Figure 6.60 is a pie chart of gender distribution among the 25 participants, excluding the ones who took the role of the offices¹.

The chart indicates that the participants are predominantly female. Because of this big difference in the sizes of samples, later analyses do not consider the gender differences.

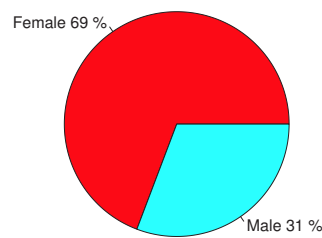


Figure 6.60: Gender-distribution of participants

Trade Types

Figure 6.61 shows the bar chart of the frequencies of the different trade types. At the experiment, 85% of the trades were issuing. 10% were circulation, and 5% were redemption. This is very different from the simulated results.

The rule of the game implied that it is more advantageous to use existing tickets. Therefore, the prediction was that the most trades would be circulation, but the prediction obviously was wrong. It seemed that many participants could not move strategically because the rule was not very clear to them. It would also be the case that circulation was felt more difficult than issuing.

Active Users

In order to grasp which age groups had the most difficulty using *i*-WAT, an analysis was done by separating the active usages (drawing and using) of

¹The genders of the participants were estimated from their photo IDs. The genders of the offices could not be estimated because their photo IDs were not taken for the game.

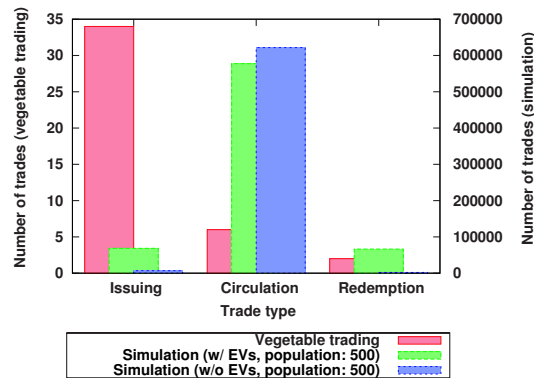


Figure 6.61: Frequencies of trade types

i-WAT into age groups of the users. Figure 6.62 (left) shows a pie chart of the age distribution.

At the first look, it seems as if tickets have been actively used predominantly by participants of ages 20s and 30s. However, this may be just that the age distribution of the participants is reflected.

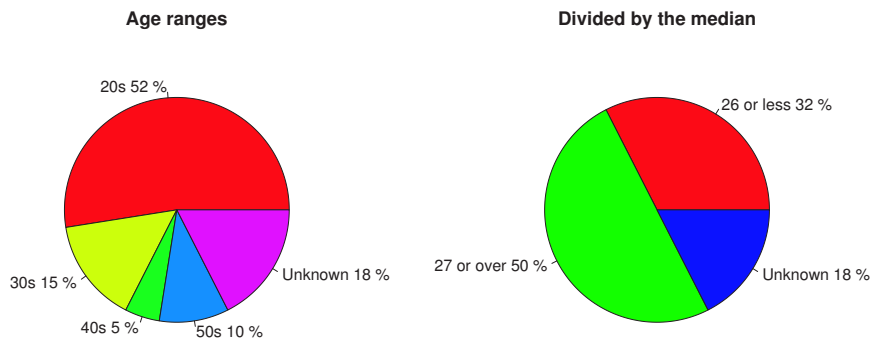


Figure 6.62: Age-distribution of ticket drawers/users

Therefore a comparison is made between participants of ages 26 or less and 27 or over, dividing the participants by the median of their ages. It resulted in the pie chart in Figure 6.62 (right).

The younger generation makes up 32% of the active usage whereas the elder generation makes up 50%. The most of unknown 18% are estimated to be in the elder generation. From these, the major active users of *i*-WAT

in this experiment was found to be the elder generation.

Passive or Non-Users

The age distribution of participants who only passively received *i*-WAT tickets or did not participate in the game at all was also investigated.

Figure 6.63 shows the age distribution (left). Notably, the only teenager among the participants is in this group. All participants of the ages 40s and 50s have actively used *i*-WAT. The unknown 33% may include elder generation, but perhaps it mostly consists of the records of unused terminals.

Figure 6.63 (right) shows a pie chart divided by the median of the ages. The elder generation makes up only 11% of the passive or non users whereas the younger generation makes up 56%.

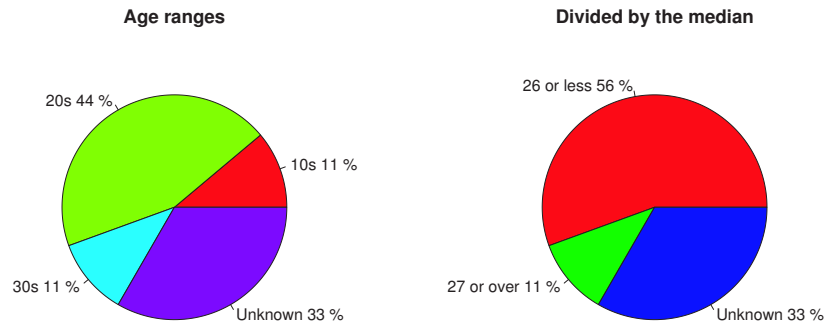


Figure 6.63: Age-distribution of passive or non-users

To see if these differences between the two generations are statistically significant, the author has applied Fisher's Exact Test to these cases, and obtained the p-value 0.07747. This may or may not be statistically significant depending on the level of required significance.

Questionnaire

The author has collected from the participants the replies to the questionnaire shown in Figure 6.64.

The collected raw data (ordinal-scale only) are shown in Table 6.6. The descriptive replies are found in section C.2.

Knowledge of the Participants As Figure 6.65 shows, the answers to the question "did you know about complementary currencies before par-

Table 6.6: Replies to the questionnaire (ordinal-scale data only)

<i>Age</i>	<i>Knowledge</i>	<i>Understanding (%)</i>	<i>Anticipation (%)</i>	<i>Usefulness</i>
19	NO	25	90	YES
20	NO	0	20	NO
21	NO	0	0	?
22	NO	0	20	?
23	NO	50	50	YES
23	NO	60	100	YES
24	NO	20	30	?
24	YES	100	10	YES
25	NO	5	30	NO
25	YES	3	3	NO
25	YES	10	20	NO
25	YES	10	10	NO
27	YES	0	0	NO
29	NO	2	3	NO
29	NO	50	50	NO
29	NO	70	30	MAYBE
29	YES	30	60	YES
34	NO	60	40	MAYBE
34	NO	60	50	NO
35	YES	50	50	YES
37	YES	25	10	YES
40	NO	20	10	NO
50	YES	20	80	YES
52	NO	0	10	?
?	NO	30	40	YES
?	NO	50	70	NO
?	YES	0	0	NO
?	YES	10	10	NO
?	YES	10	5	NO
?	YES	50	50	YES

- Q1. Did you know about complementary currencies before participating in this game?
 - Q2. How much did you understand about *i*-WAT after the game?
 - Q3. How much do you anticipate from *i*-WAT in the future?
 - Q4. Do you think complementary currencies will be useful to your life? (and why?)
 - Q5. Please tell us any suggestions or other thoughts about *i*-WAT during or after participating in this game.

Figure 6.64: Questionnaire to the participants

icipating in this game?” were 43% YES and 57% NO. In the younger generation, YES is 10 point lower than the result of all ages. In the elder generation, YES is 1 point lower. They are both lower because when the results are divided into two generations, the 6 participants whose ages are unknown are excluded from the statistics. Since these people are estimated to consist mainly of the elder generation, more elder participants must have known about complementary currencies before the game.

However, this observation may not be generalized because no test algorithm is sufficient to test samples of this small size correctly.

Understanding by the Participants Figure 6.66 shows the pie charts of the distributions of answers to the question “how much did you understand about *i*-WAT after the game?”, in which the degrees of understanding are quantized as follows: “not at all” if the degree of understanding falls into 0~24%, “a little” if 25~49%, “fairly” if 50~74%, and “very well” if 75~100%.

The percentage of participants who understood *i*-WAT at all is 46%, which is smaller than the percentage of those who did not understand it at all. The difference is made clearer by the younger generation, but the result is inverted by the elder generation.

Although this result cannot be generalized because of lack of enough samples, among the participants of this experiment, the elder generation seems to have understood *i*-WAT better.

Anticipation by the Participants Figure 6.67 shows the pie charts of the distributions of answers to the question “how much do you anticipate from *i*-WAT in the future?”, in which the degrees of anticipation

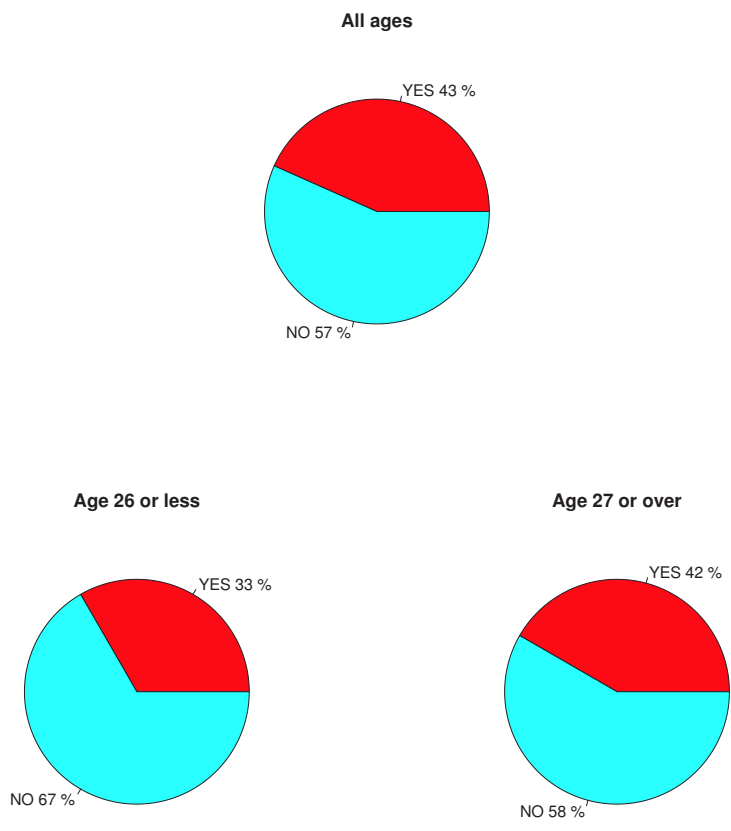


Figure 6.65: Did you know about complementary currencies?

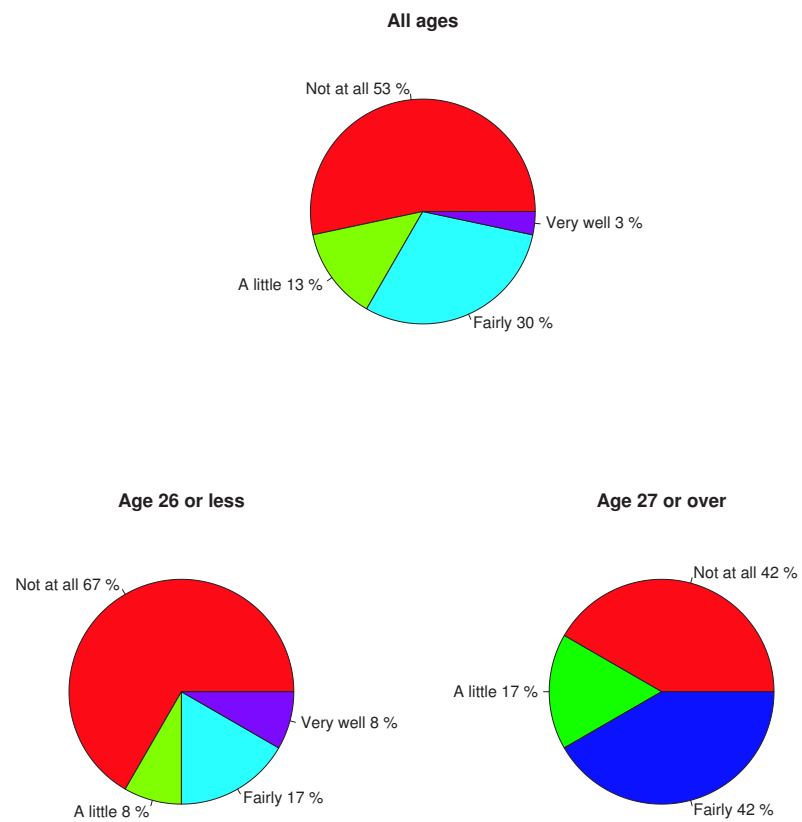


Figure 6.66: How much did you understand about *i*-WAT?

are quantized as follows: “none at all” if the degree of understanding falls into 0~24%, “a little” if 25~49%, “fairly” if 50~74%, and “very much” if 75~100%.

The percentage of participants who do not anticipate from *i*-WAT at all is 50%, which indicates that it was difficult for the participants to anticipate from *i*-WAT through the experiment. In the younger generation, there are more people who do not anticipate from *i*-WAT, whereas once again, the result is inverted by the elder generation.

Although this result cannot be generalized because of lack of enough samples, among the participants of this experiment, the elder generation seems to have anticipated from *i*-WAT more than the younger generation has.

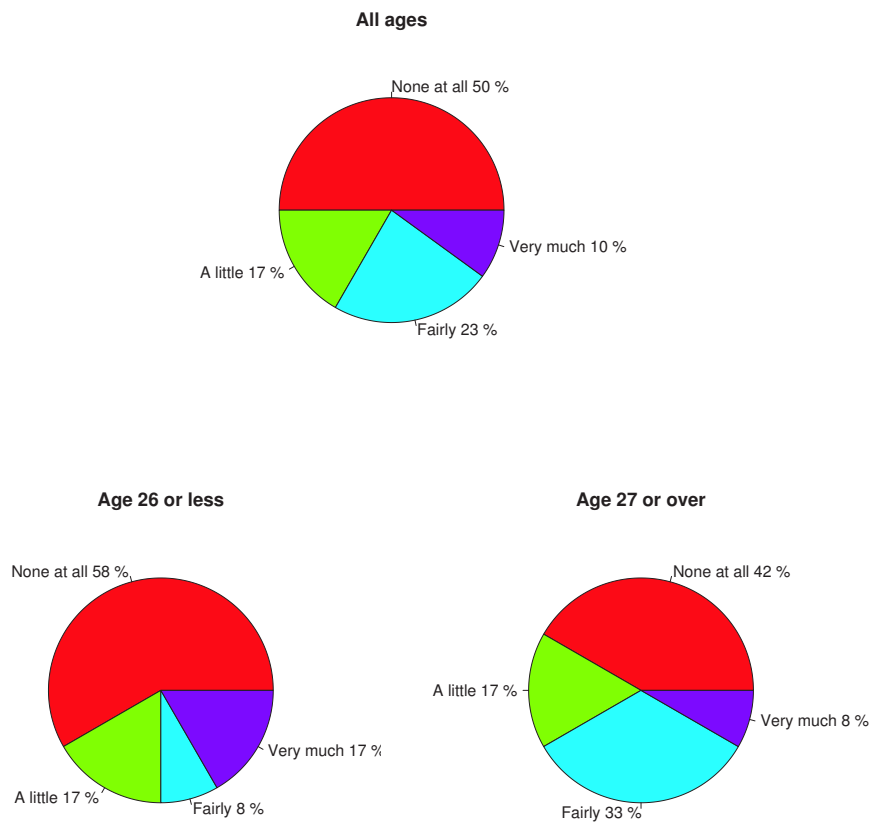


Figure 6.67: How much do you anticipate from *i*-WAT?

Judgment of Complementary Currencies by the Participants Figure 6.68 shows the pie charts of the distributions of answers to the question “do you think complementary currencies will be useful to your life?”, to which 47% of all participants answered NO, which is greater than the percentages of YES and MAYBE combined. This tendency is remained by the younger generation. However, the result is once again inverted by the elder generation, as the percentages of YES and MAYBE combined is greater than that of NO.

Although this result cannot be generalized because of lack of enough samples, among the participants of this experiment, the elder generation seems to be more open to the possibilities of complementary currencies.

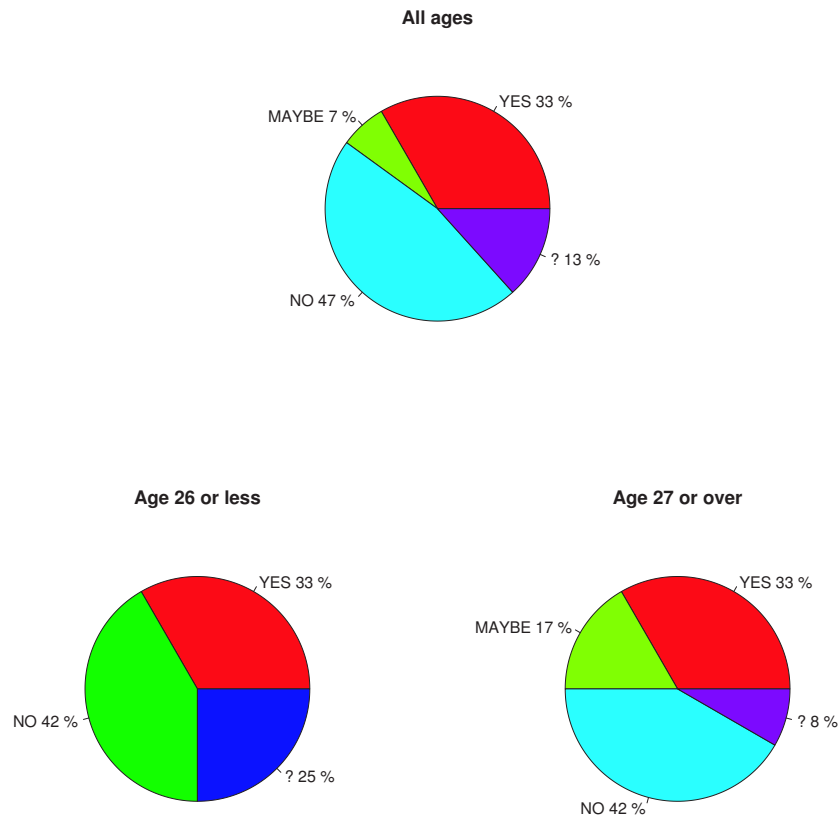


Figure 6.68: Do you think complementary currencies will be useful?

6.3 Deployment of the Reference Implementation

6.3.1 Participants and Usage

The reference implementation has already been in use by the WAT System communities. It has been used, for example, to exchange goods, such as books, with services, such as working hours for developing a free software, namely *wija* itself.

6.3.2 Statistics

Web Server Access Logs

Figure 6.69 shows the frequencies of successful downloads of *wija* during two weeks between May 20 and June 5, 2005.

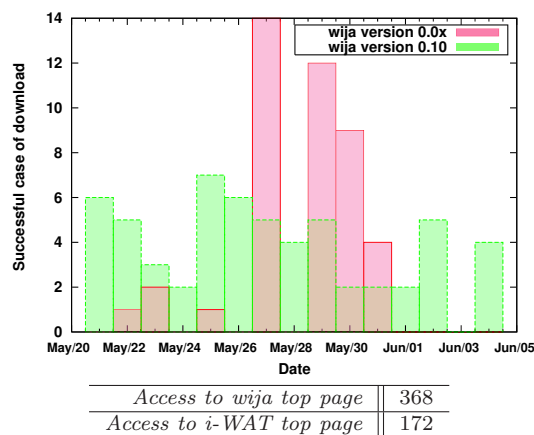


Figure 6.69: Successful downloads of *wija* May-June 2005

It was well after the release of *wija* version 0.10, but there were two to six successful downloads daily. Somehow, a notable number of downloads were made for prior versions of *wija*; perhaps this is due to some obsolete links somewhere on the web.

The number of accesses to *wija* and *i-WAT* top pages during the period was 368 and 172, respectively, showing that *i-WAT* did not attract as much attentions as *wija* did.

Figure 6.70 shows the frequencies of successful downloads of *wija* during five weeks between November 26 and December 31, 2005.

The version 0.11 of *wija* was released on the 6th of December, and the histogram shows that it was welcomed by more than 15 downloads a day for three consecutive days. But mostly, the daily downloads of *wija* is rather steadily between two and six, as suggested by these two figures.

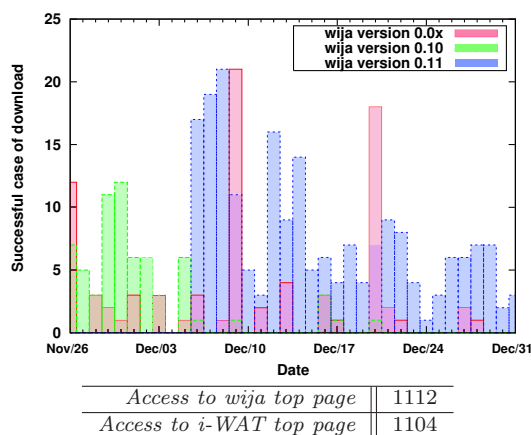


Figure 6.70: Successful downloads of *wija* November-December 2005

Figure 6.70 indicates that the switching between version 0.10 and 0.11 was successful. However, there remain mysterious downloads of prior versions of *wija*. There is no apparent correlation between the bursts of downloading prior versions and a day of week, although two instances took place on Fridays.

The number of accesses to *wija* and *i-WAT* top pages during this period was 1,112 and 1,104, respectively, showing that *i-WAT* has recently gained more attentions. Those numbers divided by 5 (weekly accesses) are 222.4 and 220.8. Compared to the weekly accesses in May~June 2005 which are 184 and 86, it seems as if both *wija* and *i-WAT* are attracting more attentions recently.

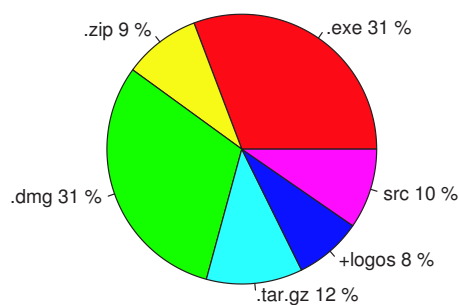
Figure 6.71 shows the number of successful downloads of *wija* version 0.11 for different platforms.

After one month from its release, *wija* version 0.11 had 260 successful downloads (as of January 14, 2006, the number of total successful downloads of *wija* version 0.11 is 332). 40% (.exe + .zip) of them were for Windows platform, followed by 31% for Mac OS X. About 10% were downloads of the source code.

XMPP Server Records

Some information can be collected from the XMPP server the author has been operating.

The number of XMPP users at media-art-online.org, which the author believes can approximate the number of *wija* users, is 202 as of December 31, 2005.



<i>wija011.exe</i>	80
<i>wija011.zip</i>	24
<i>wija011.dmg</i>	80
<i>wija011.tar.gz</i>	30
<i>wija011-logos009.{exe zip}</i>	21
<i>wija011src.tar.gz</i>	25
<i>Total</i>	260

Figure 6.71: Successful downloads of *wija* version 0.11 (\sim Jan/07/2006)

From the Author's Viewpoint

Since *i*-WAT is decentralized, no one is able to grasp the whole picture of how *i*-WAT has been used. The following is a set of information the author could collect from his own record of *i*-WAT trades:

- The number of users from the author's viewpoint:
 - The number of public keys obtained via *wija*
 - * 50 out of 73 keys in the key rings.
 - *i*-WAT traders
 - * Drawn/circulated tickets to 9 participants.
 - * Obtained tickets from 8 participants.

Reported Data

The recent versions of *i*-WAT have a functionality to obtain statistical data from the records of trades in the *i*-WAT book, which some users kindly reported to the author in April 2005. Table 6.7 shows the collected data.

The user *A* kindly reported again in January 2006, so that we can have comparative results as shown in Table 6.8.

Besides the growing usage, it shows an effect of variance over time introduced in *wija* version 0.10, by which the values of tickets need to be expressed in fractions.

Table 6.7: Statistical data from 4 participants (April 2005)

	<i>User</i>			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>Debt</i>	40kWh	65kWh	18kWh	10kWh
<i>Credit</i>	102kWh	7kWh	118kWh	0kWh
<i>Redeemed</i>	3kWh	0kWh	4kWh	0kWh
<i>Used</i>	38kWh	5kWh	4kWh	0kWh

Table 6.8: Statistical data from participant *A* (Apr 2005, Jan 2006)

	<i>Period</i>	
	<i>Apr 2005</i>	<i>Jan 2006</i>
<i>Debt</i>	40kWh	84.68kWh
		606.97yen
<i>Credit</i>	102kWh	407.59kWh
<i>Redeemed</i>	3kWh	3kWh
		100yen
<i>Used</i>	38kWh	47kWh

6.3.3 Approval Behaviors

The drawer's responsibility to approve every trade has been always seen problematic in an operational point of view. Let us see how this has been handled in practice.

The author introduces a concept of *strength of presence*, which is the frequency of being online on the observer's buddy list, quantized as follows:

- 0:** always off-line.
- 1:** mostly off-line.
- 2:** sometimes on-line.
- 3:** almost always on-line.

The author argues that actual *i*-WAT users have been utilizing this concept when they choose tickets to use.

Table 6.9 shows 8 participants the author was observing, their mean time to approval, strength of presence and the number of samples as of April 2005.

Figure 6.72 shows the relation between mean time to approval and strength of presence of those participants.

Although the number of samples is too small for a conclusion, it seems as if the participants with stronger presence tend to approve quickly the transactions involving the tickets they have drawn, and they have experienced more cases of approval than others.

Table 6.9: Participants and their approval behaviors (April 2005)

User	MTTA	Strength of presence	# of samples
A	23days, 12:00	mostly absent (1)	2
B	13days, 19:48	always absent (0)	7
C	13days, 17:24	always absent (0)	2
D	2days, 16:50	always absent (0)	4
E	1day, 15:12	sometimes present (2)	5
F	00:48	mostly present (3)	7
G	00:15	mostly present (3)	24
H	00:01	mostly absent (1)	1

MTTA: Mean Time to Approval.

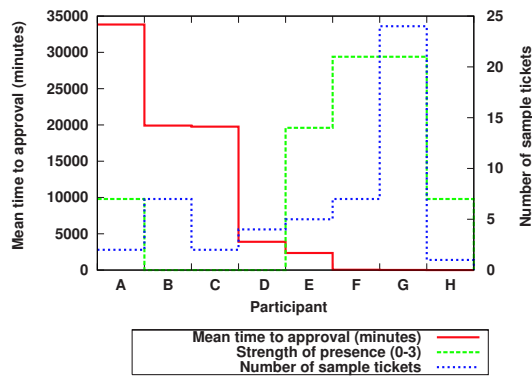


Figure 6.72: Mean time to approval and strength of presence

6.4 Cases

6.4.1 General Purchasing

Figure 6.73 shows examples of goods the author has purchased by using *i*-WAT. These are proofs that *i*-WAT is a believable medium of exchange, which has already become a reality.



Figure 6.73: Examples of purchased goods

On the left of the figure is the address sign of the author's home, and on the right are the journals of Gesell Research Society Japan.

6.4.2 Supporting by ROT

Reduction tickets have also been in use in reality. The travel expenses to and from EXPO 2005 AICHI JAPAN for the experiment of Vegetable Trading was in part paid by supporters of *i*-WAT researches across the country, in exchange for the equivalent amount of *reduction* tickets issued by the author.

This was an actualization of a theoretical model of mutual aids: the ones in need issue *reduction* tickets, so that their debts are reduced by the contributions of the supporters who accept the tickets, and supporters are helped by the utilities of the media of exchange.

The author received 26,000 Japanese yen, and issued 26 vanishing *reduction* tickets as shown in Figure 6.74 in return.

This real-life experiment also involved an actualization of another theoretical model. Since many supporters of *i*-WAT researches were not *i*-WAT users yet, they needed an exchange point who translated the author's *i*-WAT tickets into paper-based WAT tickets, whose values are reduced at the same rate as their electronic counterparts. This was a proof of concept for the translation mechanism described in section 4.11.1.



<i>Sum</i>	9.5kWh (as of December 30, 2005)
<i>Initial value</i>	10kWh
<i>Over-time rate</i>	-0.2%/week
<i>Stop value</i>	0kWh
<i>Reaches at</i>	Wed Feb 04 23:41:10 JST 2015
<i>Creation date</i>	Wed Jul 06 23:41:10 JST 2005

Figure 6.74: Example: issued *reduction* ticket

6.4.3 Barter YEN and Dollar

Among the latest experimental usage of *i-WAT* are *barter YEN* and *Dollar*, whose currency values are referred to Japanese yen and US dollar, respectively.

Figure 6.75 shows an example of *barter YEN* ticket in circulation.

The author is not yet aware of any case where *barter Dollar* is in actual use. The design of the ticket is shown in Figure 6.76.

6.5 Incidents

6.5.1 Security Incidents

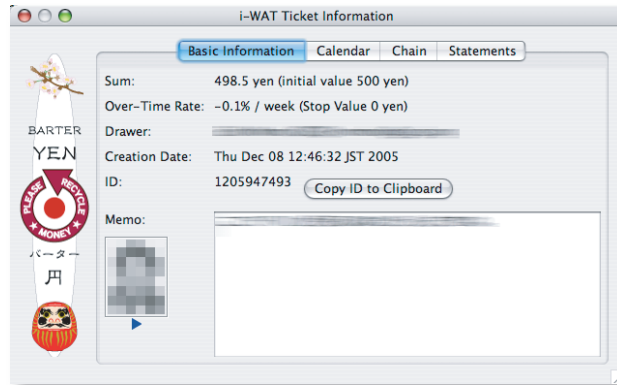
Overview

We have experienced one actual case of a security incident: a drawer lost his secret key, along with the whole data of transactions with which he had been involved. This happened because of his misunderstanding that all *i-WAT*-related data were stored in a server. Based on this misbelief, he erased his disk completely.

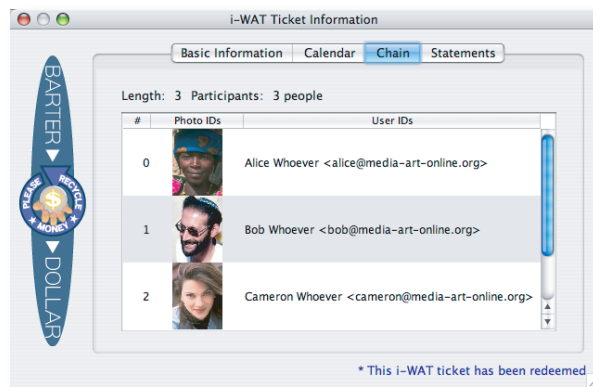
We have successfully recovered from this incident.

Recovering Data

As noted before, in *i-WAT*, a transaction is recorded by up to three parties (because circulation involves a user, recipient and the drawer), so the data



<i>Sum</i>	498.5 yen (as of December 30, 2005)
<i>Initial value</i>	500 yen
<i>Over-time rate</i>	-0.1%/week
<i>Stop value</i>	0 yen
<i>Reaches at</i>	Thu Feb 06 12:46:32 JST 2025
<i>Creation date</i>	Thu Dec 08 12:46:32 JST 2005

Figure 6.75: Example: *barter YEN* in circulationFigure 6.76: The design of *barter Dollar*

is naturally replicated, which can be used for recovery from crash accidents.

Although automation of this procedure is yet to be implemented, the author could successfully recover the data of *i*-WAT tickets in his *i*-WAT book from the records of his trade partners.

Recovering a Secret Key

It is not possible to recover a lost secret key as it was. However, as it has been noted several times already, the identifier in *i*-WAT is the user ID of a public key instead of its hash value.

It has been confirmed that the revived *i*-WAT ticket he has drawn became usable again after his creation of a new key pair with the same user ID, and rebuilding the web of trust around him.

6.5.2 Availability Incidents

Unavailable Proxy Service

The proxy *proxy@media-art-online.org* was sometimes unavailable to provide the service because its port seemed to have been inaccessible from outside.

After some time, the author changed the port the proxy uses for the service, which is certain to be accessible. This change did not affect other clients at all when using the proxy, because the port number and the IP address of the proxy are transferred to other clients through the protocol of service discovery.

The author believes that this has proven the excellence of Jabber/XMPP and related protocols in the ability to rendezvous.

Unavailable XMPP Server

Although it has some good properties, Jabber/XMPP is still problematic in its availability of services.

The XMPP server *media-art-online.org* was accidentally down for a few number of times because of a bug in *jabberd 2* which did not handle removal of users well (the current version works fine). There have been planned power outage once a year, during which the server is not accessible.

Although the author believes that these incidents have not distracted users, as they are not yet so depending on *wija*, but there is a necessity for moving toward a more P2P-oriented messaging system, which does not necessarily become unavailable when a part of the system goes off-line.

Chapter 7

Discussion

7.1 Meaning of the Results

7.1.1 Non-Intuitive Consequences

We have seen a series of consequences that are not intuitive in the simulations.

Growing the number of whitewashers in an MCS has a paradoxical effect of increased welfare of the whole, especially when mass-market partnership is applied. Such systems may go into an unhealthy state without members noticing it. A growing number of existing complementary currencies are going for electronization, which will have an effect of broadening their markets. Designs of many peer-to-peer currencies which are variants of MCS are concerned with scalability. Those may be based on naive observations that scaling up the systems will help. We would suggest that at least some punishment mechanisms against whitewashing should be introduced in such systems.

Avoidance of risks by stretching the chains of endorsements (EV2) in (presumably) both WAT and *i*-WAT has an effect of suppressing trades. This evasive action needs to be accompanied by other actions which aim to eliminate tickets or to increase the chances of doing so.

Our optimization for use of variance-over-time tickets assumes graceful thinking of participants when they accept *reduction* tickets, because they must agree to contribute to reducing the debts of the issuers. Yet the tickets have effects of accelerating trades, reducing the occurrences of bankruptcies, and as a result, increasing welfare of the whole. On the other hand, the same optimization assumes just self-interest when applied to *multiplication* tickets. However, the tickets have effects of suppressing trades and limiting welfare of the whole.

While usage of *reduction* tickets is encouraged for stability of the system and welfare of all, usage of *multiplication* tickets are advised to be minimal.

7.1.2 Moral Hazards and Counteraction

A part of this work can be seen as a computational confirmation to what R. Douthwaite said on the subject of LETS, a form of an MCS, in an interview[28]: "... the whole weakness of a LETS system is that nobody ever specifies how quickly you're going to repay that debt back to the group and nobody ever chases you." There is no doubt that such systems encourage moral hazards.

WAT/*i*-WAT do not specify the timing of redemption either, but two out of three evasive actions for *regular* tickets investigated in this dissertation have a direct effect of accelerating redemptions. This acceleration is spontaneously enforced out of self-interest of the participants; they are interested in eliminating tickets with which they have been involved so that they can reduce risks imposed by the security rule.

There is a possibility that EV2 may arouse moral hazards, because when the chain of endorsements is long enough for the presented ticket, the receiver can be indifferent to the risks, and may accept it without taking enough precautions. However, misbehaviors such as forging is always a possibility as [78] suggests, whose risks are never reduced even if the chain is lengthened. Receivers are always at the edge of the chain of responsibilities where all consequences must end.

7.1.3 NEO and Motivations for Creations

An important question is whether or not making information freely distributable deprives their producers of motivations to produce the information.

The author argues that NEO can motivate people to produce information because then they will be allowed to issue *reduction* tickets. These tickets can be issued only when the lenders agree to take it, and rational lenders would take them only when those tickets are likely to be taken by others, because they do not want to hold on to tickets whose values keep decreasing. Which suggests that producers of information will be rewarded by different levels of freedom to issue *reduction* tickets based on the popularities of their work. It may provide producers of creative works with reasonable career strategies, even though their products are freely shared.

We will investigate how such different levels of freedom will affect the exchange systems.

7.1.4 Implied Institutional Changes

Every new technology must be supported by social changes when it is deployed in real life.

[1] describes this as a relationship between *architecture*, or the matrix of concepts designed into a technology, and *institution*, or the matrix of con-

cepts that organizes languages, rules, job titles and other social categories in a given sector of society. It has explored the tension in the deployment of P2P, between the engineering story of rationally distributed computation and the political story of institutional changes through decentralized architectures.

As a P2P medium of exchange within P2P systems, *i*-WAT implies needs for similar institutional changes, which may need to be tackled independently from the architectural design of *i*-WAT.

Enforcement of the security rule, for example, requires conformity to a norm which may be resulted from a rather centralized effort of conversion. Finding a right way of constructing an institution may require an equal amount of work as those have been put toward designing the architecture.

One thing we may consider is to utilize institutions that are already there, which have already gone through necessary changes to support decentralized architectures. The author's suggestions include open source/free software communities which are decentralized in nature, but most of their participants conform to a norm. The author is interested in the emergence of such a norm.

7.1.5 Differences between Models and Practices

Numerous factors are ignored in the simulation model for simplification.

In particular, the model assumes an existence of a mechanism such that the extent of debt owed by a drawer can be monitored by the prospective lenders to enforce the limit on issuing tickets. This mechanism needs to be implemented in a decentralized way in order not to obstruct the autonomy of the system.

A distributed query method to achieve this, which is an application of *fair-sharing* protocol by Ngan et al.[61], has been discussed. Efforts will be paid for realization and installation of the scheme.

There should also be a concern about the cost of always adding the drawer of the ticket to one's acquaintances when receiving a ticket. We believe there are many measures to be taken to reduce such a cost, including facilitation for public-key exchanges. Some such facilities have already been implemented in the reference implementation.

7.1.6 Gender and Age-Distributions of Users

It was found that voluntary participants of the Vegetable Trading experiment were predominantly female¹. It was also found that elder participants (age 27 and over) could use *i*-WAT better than the younger ones (age 26

¹This may be attributed to the fact that the experiment was conducted on a weekday, and more women than men are either unemployed or self-employed, which made female participants more easily attend the experiment.

and less), and the answers to the questionnaire also supported this observation by indicating that elder ones understood *i*-WAT better, and had more anticipation to the complementary currencies in general, including *i*-WAT.

Let the author investigate to see if these are of any significance.

It is apparent from the lack of enough samples that these results alone are not statistically significant. But the observation on age groups coincides with the author's personal experiences through the development and daily use of *wija* and *i*-WAT (most feedbacks to these software are from members of Gesell Research Society Japan, who are predominantly in the elder generation).

Moreover, existing studies on the user population of LETS suggest characteristics of typical users of complementary currencies, which match the results from the Vegetable Trading experiment to some extent.

[39] is a study on 8 groups of LETS in Australia, which states that their users are predominantly female. [109] is a study in UK, which is an analysis of responses from 810 people out of 2515 postal questionnaires they sent out. It states that users of LETS in UK are predominantly aged 30-49, women in relatively low income groups and those who are either not employed or are self-employed. [37] is an overview of a LETS in Korea. It states that the age distribution of its 590 members is in the order of 30s, 40s, 20s, 50s, 60s and 10s.

If the observed differences are real for the 30 volunteers in the Vegetable Trading experiment, however, there are still some doubts that these are pseudo-correlation with some third variables.

There are two hypothesis on the observation on the age groups:

- Hypothesis 1: general attitude towards people

The younger generation may see things overly critically, but the elder generation tries to accept assertions before starting critical thinking.

- Hypothesis 2: self respect

The younger generation is not hesitant in admitting that they do not understand something. The elder generation does not want to admit their not understanding something.

These hypotheses have some common problems in light of the conditions of the experiment. *i*-WAT is at the moment a technology that requires complex operations. The younger generation seems to be understanding and using equally complex cellular phones better than the elder generation does. Contrary to hypothesis 1, the elder generation is more critical of cellular phones, and contrary to hypothesis 2, they do not hesitate admitting that they do not understand how to use versatile cellular phones.

Hypothesis 2 seems to be also refutable by the fact that the elder generation could use *i*-WAT better – the observation possibly statistically significant depending on the required level of significance.

Still, it may be that the effects as explained by hypothesis 1 made the elder generation more actively participate in the experiment, which resulted in more efforts to use *i*-WAT, which led to better understanding of it, and then more anticipation came from the understanding. But this only explains a mechanism how the elder generation can be tempted to use, understand and anticipate from this technology, which we may just want to utilize.

While verification is certainly a necessity, by experimenting with increased number of participants, for the time being, the following needs to be considered. Deployment should assume that at least initial users are from the elder generation, possibly predominantly female, and so the human interface needs to be re-designed. At the same time, efforts are required to investigate how to appeal to the younger generation.

7.1.7 Implications of the Experimental Outcomes

Through the Vegetable Trading experiment, the author proved that people can establish electronic trade relations out of the state of no network infrastructure and no trust over public keys of participants.

The success of this experiment implies that we can trade things electronically by establishing an appropriate level of trust without relying on an outside authority, even without a network infrastructure.

This, as the experiment symbolizes, can be applied to many situations such as mutual help among survivors of catastrophic events, trading storage spaces or CPU time in P2P systems, and so on, where self-organization is required. Self-organization implies spontaneous start, therefore those who have goods or skills that are not utilized can give them to those who need them at will, improving the utilities of excesses which we might possess.

However, as the responses to the questionnaire suggest, more work is required as to improving the human machine interface of the system and educating general public on public key cryptography and WAT/*i*-WAT.

7.2 Related Work

7.2.1 Overview

There have been a number of researches and experimental or practical implementations of P2P currencies. Many of them raise issues of dependability, but only in the sense of availability of the service. Little of them have been concerned with economic models.

7.2.2 Centralized Debt-oriented Currencies

All of the payment systems discussed in this section are distributed, but their models of economies are centralized. They may be subject to the problems

of an MCS as described earlier.

MojoNation[6] was a system of file sharing (regrettably it does not exist anymore), which is perhaps the most well-known in the context of P2P currencies. Every transaction in the system required payment of *Mojo*, which was controlled by a single token server. MojoNation's economy was a form of an MCS.

Karma[98] is a distributed payment system based on *bank-sets*. It is designed to work on DHTs (Distributed Hash Tables) such as Pastry[67]. Its trust model is based on the secure routing discussed in [12] which relies on a set of trusted certificate authorities. Since the whole concern is consistent management of an accounting system in a distributed environment, Karma can be seen as a form of an MCS.

Ripple[26] is a distributed payment system. It finds a chain of credit connections between parties to make payments. If *A* and *B* are mutually trusted, and *B* and *C* are mutually trusted, and if *A* wants to make a payment to *C*, then *A* pays (owes) to *B* so that *B* pays (owes) to *C*. The account information is made private among trusted parties. Ripple is a form of multiple MCS, in which several MCS's mutually have accounts for one another.

7.2.3 Decentralized Debt-oriented Currencies

Samsara[16] is a fair P2P storage infrastructure in which each peer that requests storage of another must agree to hold a *claim*, or incompressible space, in proportion to their consumption. Claims can be forwarded among chains of nodes, eliminating themselves when cycles are found; this is somewhat similar to redemptions of WAT/*i*-WAT tickets.

Geek Credit[44] is a currency system that is close to *i*-WAT. It defines *Geek Credit policy*, which is similar to the *i*-WAT state machine, but the problem of double-spending is handled differently. Geek Credit detects double-spending at redemption, so that each transaction does not need to be consulted with the drawer. While this simplifies the protocol, exploiting this system may be easy, and enhancement to the effects of evasive actions as witnessed in the simulations of *i*-WAT is not to be expected.

PPay[111] is another currency system that is close to *i*-WAT. PPay handles the problem of double-spending in almost the same way as *i*-WAT does; it requires approval (process of *reassignment*) by the issuer of the coins when they are transferred to other parties. The difference is that this authority is duplicated in PPay. It assumes that an external banking facility exists, which exchanges the governments' fiat money with digital coins. Such facility may be given authority to reassign coins.

This makes the currency more available, but it also makes the protocol more complicated than that of *i*-WAT. The author believes that availability of the issuers can be increased by applying existing fault-tolerance

techniques, independently from the currency design. Since it enhances the utility of tickets, it may increase their levels of freedom to create a medium of exchange. Therefore, some issuers may find it beneficial to pay the cost of applying such techniques.

7.2.4 Decentralized Labor-oriented Currencies

BitTorrent[15] is a file distribution system on a tit-for-tat basis. Each peer is responsible for attempting to maximize its own download rate. If peers do not cooperate they *choke* their partners, or temporarily refuse to upload.

7.2.5 Incentive Techniques

Fair sharing protocol by Ngan et al.[61] is a distributed query algorithm which can detect lies about participant's debts probabilistically, which the author believes is broadly applicable to many problems.

Feldman et al.[22] investigates some incentive techniques to tackle the problem of free-riding. They found that when penalty is imposed on all newcomers, the system performance degrades significantly only when the turnover rate among users is high. This observation is applied to the study on this dissertation in such a way that all participants start with relatively small number of acquaintances when using small-world partnership, which can work as a penalty to whitewashing.

Stamp trading[53] is a generalized protocol for reputation and payment. This is a theoretical framework, and no practical issues such as double-spending have been addressed.

7.3 Comparison of Trust Models

7.3.1 Comparison with Geek Credit/Ripple Trust Model

Overview

The trust models (definitions of mutually validating relation $\overset{v}{\leftrightarrow}$) of Geek Credit and Ripple should essentially be one and the same, which is illustrated as Figure 7.1. While it simplifies the protocols, this model implies that these currency systems are more vulnerable to strategies than *i*-WAT is.

Strategy-Resistance

In Figure 7.1, if Alice is the issuer of a Geek Credit ticket, then anyone from Bob to Ellie can double-spend the ticket for a trade with anyone not listed in the chain of trust, and get away from being detected until later. Even upon detection, the hazard may have been caused by a total stranger to Alice, which may make enforcement of fairness difficult.

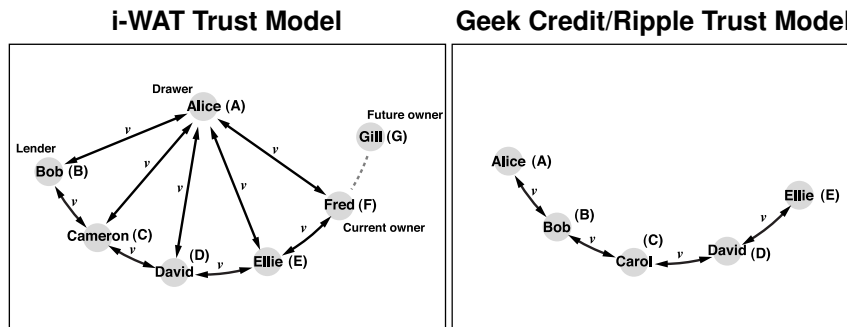


Figure 7.1: *i*-WAT and Geek Credit/Ripple trust models

If Alice wants to make a payment to Ellie using Ripple, then it is possible that Alice is imaginary, and was created by Bob, who is the actual purchaser of goods or services from Ellie. Bob may be able to spend as much as he wants using his imaginary friend, which might not be detected until very later.

The author believes that the safety of Ripple can be improved if it gives up protection of privacy to some extent; then incentive mechanisms similar to those of *i*-WAT may be able to be applied to its design.

Idempotence

In the Geek Credit/Ripple trust model, it requires a storage space to hold every exchange medium a peer has received if double-spending is to be detected by participants upon usage instead of at redemption. However, double-spending is to increase the drawer's debt, and it has no apparent effect on the receiver's benefit. Participants do not have incentives to spend storage space for the benefit of the drawers.

Autonomy

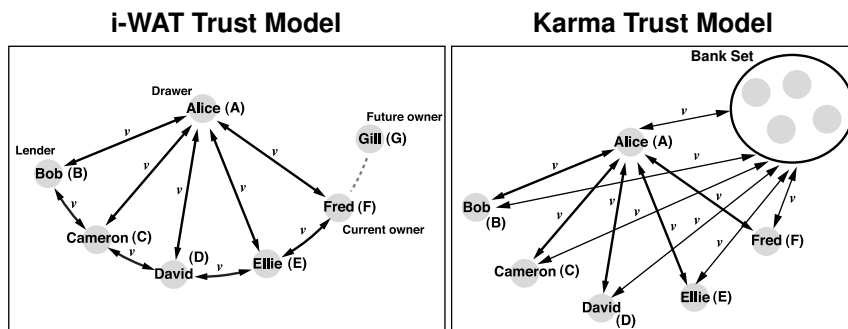
The Geek Credit/Ripple trust model provides a higher level of autonomy because it requires less, but it is not backed up by strategy-resistance.

7.3.2 Comparison with Karma Trust Model

Overview

The trust model of Karma is illustrated in Figure 7.2. It shows the case in which Alice and all other participants are trader partners.

This model requires an absolute trust on the bank set which maintains the accounts of participants. For this reason, this model is essentially the same with those of any currencies based on an MCS, including MojoNation.

Figure 7.2: *i*-WAT and Karma trust models

Strategy-Resistance

Because Karma is dependent on a set of trusted certificate authorities, it is easy to be an impostor once those authorities can be fooled. The participants do not have an incentive to check the identities of partners themselves because it is usually taken care of by the authorities: this is a typical case of a moral hazard.

As described before, it is not clear who takes responsibility of the debt if, for example, Alice leaves the system with a negative balance on her account (presumably everyone else does).

Idempotence

In the Karma trust model, complex messaging is required to detect double-spending especially because their bank accounts are managed in a distributed way.

Autonomy

The Karma trust model provides a limited level of autonomy, because emergence of the system requires the construction of the bank set.

Chapter 8

Conclusions

P2P barter currencies can be powerful tools for promoting collaborations and building sustainable relationships on the Internet. *i*-WAT is a proposed such currency based on the WAT System, a polycentric barter currency using WAT tickets whose values are supported by chains of trust.

The author believes that an architecture for forming and maintaining relationships of collaboration has been constructed, through which individuals can take part in generation of desired autonomous, distributed and cooperative mechanisms over the Internet. This architecture realizes the following:

1. Producers of free information such as free software can be supported by peers, being allowed to issue *reduction* tickets to obtain goods or services they need. Being drawers, they can monitor the happenings of trades using their tickets as a proof of their being supported by the communities, which can motivate such people to produce new information.
2. A stable system of exchange, in which there is little or no incentive for the participants to escape from their responsibilities, can be constructed by applying reduction-over-time feature to the WAT System, which has a circular structure of responsibility.
3. Even when a network infrastructure is destroyed, and there is no trust in the authenticities of public keys of participants, electronic trades are made possible by the participants' spontaneously constructing appropriate level of trust.

Achieved Autonomy In this research, we clarified the *i*-WAT trust model. To implement the model by dynamically building an appropriate web of trust, we showed that it would suffice if the behaviors of participants satisfy the following three properties:

1. *mutual signing by knowing*
2. *mutual signing by participation*
3. *mutual full trust by participation*

Likelihood of satisfaction of these properties is supported by the (dis)incentives imposed by the semantics of *i*-WAT.

This spinning of webs of trust was witnessed in an experiment conducted at EXPO 2005 AICHI JAPAN using wireless communication devices. The experiment proved that people can establish electronic trade relations out of the state of no network infrastructure and no trust over public keys of participants.

Achieved Safety This research investigated an extension to the design of *i*-WAT to implement ROT (Reduction Over Time), which has potential effects of both promoting exchanges and providing participants with means to mutually support peers by sharing debts among one another as a form of currency. The extended design is shown to be incentive-compatible by a game-theoretical analysis. In particular, we predicted that the following properties will hold:

1. *Rapid circulation*, or a *reduction* ticket will typically circulate at high speed until its effective value reaches the scheduled minimum, and
2. *Vanishment equilibrium*, or the system will be most stable if the values of tickets are to be reduced down to zero.

This research also investigated an extension to the design of *i*-WAT to implement MOT (Multiplication Over Time), which can help participants in P2P systems who are in strong need of some specific resources. The extended design is shown to be incentive-compatible by a game-theoretical analysis. In particular, we predicted that the following properties will hold:

1. *Deferred redemption*, or if the lender accepts such a ticket, they are likely to use it against the drawer themselves, and to defer it until the effective value reaches its maximum.
2. *No strategic default*, or the drawer is incentivized to successfully redeem such a ticket.
3. *Acceptance criterion*, or the drawer can only increase the change of acceptance of such a ticket by signaling how they are unlikely to default.
4. *Ease of Flow*, or if the lender is willing to take the risk, later participants are likely to accept the ticket as the chain of endorsements grows.

This dissertation quantitatively compared by simulations the effects of core designs of MCS, WAT and *i*-WAT currency systems in the presence of whitewashers. It also investigated the effects of *reduction* and *multiplication* tickets in *i*-WAT.

We showed that the design of MCS is especially problematic in the case where participants can freely choose their trade partners from the whole population. Although choosing partners within one's own network of acquaintances has an effect of punishing those who try to exploit the system by re-joining, other users may remain indifferent to the growth of such bad users, possibly encouraging moral hazards.

The security rule of the WAT/*i*-WAT makes the risks apparent to the participants, motivating them for evasive actions out of self-interest which has positive effects to the community of peers. We showed that linkage with drawers, introduced to *i*-WAT for detection of double-spending, enhances the effects of such evasive actions. Simulations demonstrated that *reduction* tickets can benefit welfare of the whole even though accepting such a ticket requires a graceful thinking instead of self-interest. Those are what the author believes on the brighter side of risks in P2P barter relationships.

On the darker side, the author has found that *multiplication* tickets limit welfare of the whole when those who accept such a ticket tries to maximize their gains and minimize their losses. This is the only case the author has discovered so far in the design of *i*-WAT where pursuit of self-interest fails.

Through an actual security incident, the author has proved that a participant can recover from losses of their secret keys and *i*-WAT ticket data, with help from other participants.

Achieved Integration The author has developed an Jabber/XMPP client called *wija* in order to put *i*-WAT into practical use. The author has been experimenting on user interfaces for exchanging public keys and on automated trust settings, so that participants can satisfy the *i*-WAT trust model with little or no subjective communication cost.

Extensibility of the *i*-WAT protocol has been tested when the author added variance-over-time features such as ROT and MOT to the existing protocol.

The translation mechanism with other currencies, as well as the theoretical model of mutual aids with *reduction* tickets, have also been tested when this research was supported for the experiment at EXPO 2005 AICHI JAPAN.

For the counteractions against moral hazards to work, prospective lenders need to know the extent of the debts of participants. Efforts will be paid for design and installation of an operable distributed accounting mechanism using which participants can share information about tickets issued by others in circulation.

Chapter 9

Recommendations

9.1 Research Plans

Above all, the author would like to make *i*-WAT more usable to the general public by reworking its human interface and by promoting the concepts of public key cryptography and barter currencies.

The author will try to verify further the results of this work as we continue to accumulate more experiences in using *i*-WAT. In particular, the author would like to measure the cost of trust which is an important factor in the designs of protections against strategies.

There is a difficulty in investigating how *i*-WAT has actually been used, because of its totally decentralized nature. To overcome this difficulty, the author would like to propose an introduction of a *diplomatware* to the reference implementation of *i*-WAT.

Unlike a *spyware*[108], which is designed to intercept or take partial control of a computer's operation without the *informed consent* of the user of the computer, a *diplomatware* collects information regarding the activities of the user of the computer with the consent of that user.

A future version of *i*-WAT software will be equipped with such a *diplomatware*, whose functionality can be turned off by the user, which collects information of *i*-WAT trades, and reports it to some web page in an anonymous form. The gathered information will be made available to the general public (in perhaps a graphical way) so that everyone can share the idea of how *i*-WAT is spreading in the world.

9.2 Unimplemented Features

9.2.1 Distributed Auditing

The author intends to implement features to the software for monitoring excessive issuing: sharing information about tickets issued by others in cir-

ulation. The author believes that this can be done in a decentralized and trusted way by the protocol described in section 4.8. The author will investigate how we can limit the exposure of information to protect privacy of participants.

9.2.2 Privacy Support

Optional encryption will be introduced to the *i*-WAT protocol.

Out of a privacy concern, in the current *i*-WAT protocol, the item for a trade is omitted in `<accept/>` message sent to the drawer. This implies that it is also omitted in `<approve/>` message sent back from the drawer, which makes the message information less manageable than it should be in the *i*-WAT book. Also, the difference between the `<accept/>` messages sent to the drawers and those sent to the users makes using multicast impossible even when trades are processed over wireless communication channels.

One possible solution to this is encrypting the name of the item with the public keys of direct trade partners, so that the drawer cannot know what it is.

An alternative is omitting the item for a trade all together from the `<accept/>` messages, and let the clients retrieve the name of the item from the record of the corresponding `<use/>` or `<draw/>` message.

9.2.3 More Public Key Management Support

Automated trust settings, or automatically setting full trust on the owners of public keys with whom the user has exchanged *i*-WAT messages for a successful trade, has proved to be useful by the Vegetable Trading experiment. This feature will be implemented in the publicly available version of *i*-WAT.

Other public key management support features, such as facilitating public key exchanges with newcomers, or automatically updating the copies of a public key stored in adjacent clients if a user has signed a key, will be implemented and tested.

9.2.4 Enforcement of the Security Rule

Although enforcement of the security rule requires social approaches, its implementation at the data representation level is a necessary part of its building block, and it will be implemented in the near future version of *i*-WAT.

9.2.5 Automatic Backup and Distributed Restore

The feature of automatic backup will be introduced for the safety of the data. The recovery process using the data collected from the past trade partners will also be automated.

9.3 Application: Distributed Computing

As an example of a concrete application of *i*-WAT, the author would like to implement a (demonstrative) system of distributed computing over *wija*, in which computing resources are traded using *i*-WAT.

A candidate for the subject of computing is a simulation of *i*-WAT itself. The author has already discovered that simulations involving variance-over-time requires much more computing resources than those with regular tickets only. A simulation of how *i*-WAT can be used, over the people's network of *wija*, can also be an advertising tool for the further deployment of *i*-WAT. The author would like to think of a highly graphical way of representing the simulation, perhaps involving some animated characters, so that it can attract many people.

9.4 Application: Sharing Creative Works

As another example of a concrete application of *i*-WAT, the author would like to implement a system of sharing creative works, such as free software, music, and educational materials.

The basic concept is as section 4.12 describes: creative works are freely shared, while their producers are supported by the peers, being allowed to issue *reduction* tickets to satisfy their needs. The author would like to create different sets of on-line communities, each for sharing a specific type of works such as software, music or educational materials, consisting of producers of such works who create their works based on those of others, providers of resources for creation (such as computing power) and audience.

9.5 Application: Post-Catastrophic Recovery

As yet another example of a concrete application of *i*-WAT, the author would like to propose a way of mutual help in post-catastrophic situations, which the author devised with Eiichi Morino and members of Gesell Research Society Japan.

9.5.1 Principles and Requirements

Principles

We do not aid people by collecting money and sending it to the affected places. Instead, we will build a mechanism so that anyone in the world can transfer funds to someone in an affected place.

We will build a model so that in a long term, the local economy will stand independently. The major actors are not us, but people in the affected places.

Requirements

We hear a lot about donations not reaching the very people who need them. We need to design a way in which the funds are transferred safely to the intended destinations.

We need to avoid helping to build societies so heavily in debt after efforts for reconstruction. We need to avoid helping to build societies dependent on aides from others.

Complementary currencies may help installing self-sustainable economy in the disaster-affected places. But if those currencies require central authorities, they will be costly in their operations. We are not the ones who need funds, but the people in the affected places are. We need to design a model which requires the smallest overhead as possible.

9.5.2 WAT/*i*-WAT for Post-Catastrophic Recovery

An Example to Think About

Let us think about a desirable form of aiding people in the post-catastrophe places based on the above principles and the requirements. Let us think about an example of a fishing village.

1. Suppose that the villagers lost the instruments for fishing including ships.
2. Suppose that the villagers lost the infrastructure to support the village.
3. Suppose that there is an NGO (Non-Government Organization) placed in the village or its vicinity to help recovery of the village.

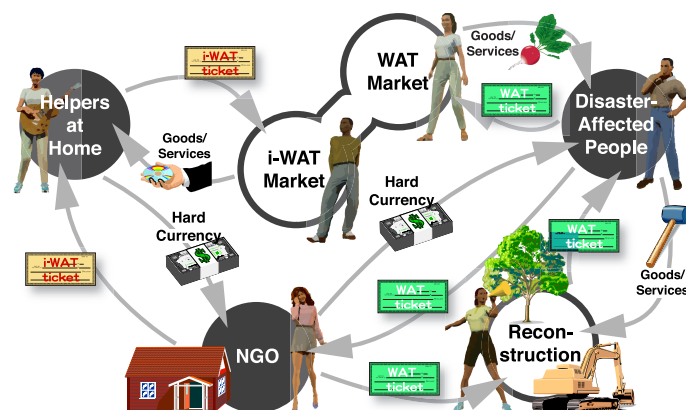
The Model

The model is illustrated in Figure 9.1. It is designed in the hope that everyone in the world can help each other as peers.

Usage of WAT in the Disaster-Affected Village In the disaster-affected village, the local people will work for the local people.

The villagers acquire fishing instruments by issuing WAT tickets. By issuing the tickets, the villagers promise that they will provide some goods or services when the tickets return to them in the future. They can possibly issue (vanishing) *reduction* tickets so that their debts will be reduced over time by the contributions from those who accept the tickets.

In case the merchants do not accept WAT tickets, the villagers can give the tickets to the NGO to receive funds in return. By knowing that there is a way to exchange WAT tickets with money, it will become easier for people to accept WAT tickets.



- WAT and *i*-WAT markets are connected by entities which issues one type of tickets by promising another. The NGO in the diagram is an example of such entities.

Figure 9.1: A model of WAT/*i*-WAT for post-catastrophic recovery

The NGO, by using the WAT tickets they have obtained, can employ people in the village and its vicinities to work for restoring the infrastructure.

Usage of *i*-WAT in the Rest of the World The NGO issues *i*-WAT tickets which can be purchased on the Internet by the rest of the world.

Those *i*-WAT tickets promise to give WAT tickets issued by the villagers (therefore if such a WAT ticket is a *reduction* ticket, the corresponding *i*-WAT ticket is also a *reduction* ticket with the same reduction rate). If someone in the rest of the world purchases one of those *i*-WAT tickets, it means that he or she helps some particular villager.

The NGO purchases the WAT tickets issued by the villagers using the money the organization obtained in exchange with their *i*-WAT tickets.

Expected Consequences

The village in some day will be reconstructed. The villagers will pay back to the rest of the world with their working and its results.

The helpers in the rest of the world can use the *i*-WAT tickets they have purchased for trades on the Internet. Or they can donate the tickets to the NGO (or in fact, to anyone) if they wish. Or they can choose to keep connected with the village; they can save the tickets, and exchange them with the real WAT tickets issued by the villagers after recovery.

In this way, WAT/*i*-WAT connect the village and the rest of the world.

If people feel that the NGO is an overhead, they can naturally invent ways to get around it. Gradually, the NGO will complete its role as a

medium for spreading the idea of WAT/*i*-WAT. It is the birth of a new economy where everyone can participate spontaneously. It will be self-sustainable and open at the same time.

Strong Security for WAT → Hard Currency Exchange The money you will pay in exchange with *i*-WAT tickets will be transferred to people in the affected area by way of the NGO. We will install a mechanism so that the money is transferred safely and with certainty. In any case, the money is only transferred in exchange with some WAT tickets issued and presented by the people there.

Donating *i*-WAT Tickets Instead of Hard Currency If you are hurt, but do not feel as if you are rich enough to purchase a WAT ticket having a value of 1kWh, which costs 1 US dollar or 100 yen, then you can issue your own *i*-WAT ticket which has a value of 1kWh, and donate it to the NGO. Which means that you promise to pay back by approximately 6 minutes of working in some future.

The mechanism of WAT/*i*-WAT allows you to help people in the disaster-affected places by either of the following ways:

1. purchasing WAT tickets with hard currencies like US dollar or yen, or
2. donating *i*-WAT tickets promising your own labors in the future.

9.5.3 Implementation of the Proposed Model

Required Computing Resources

Our reference implementation should be able to handle actual transactions based on the proposal. *wijabot* will be useful for automating the procedures of issuing *i*-WAT tickets and approving transactions involving such tickets. The NGO just needs to have a computer with moderate performance and IP connectivity with moderate bandwidth.

How We Should Proceed

There is no doubt that installation of such a social program requires help from many enthusiastic people, especially while there is no precedents. Spreading the idea that barter currencies can help the situations can be helpful, and we should look for people who are willing to spend time installing and operating the social program with us.

The author has already made a proposal at ccTsunami[13].

Afterword

Capt. Jean-Luc Picard must have been right in stating that the economics of the 21st and 24th centuries are different, and acquisition of wealth still seems to be the driving force in our lives. The author suspects that this is so because people tend to invest on things that last long, and money in the present form is designed to last long.

If money does not exist, or if it deteriorates or disappears appropriately, then, the author imagines, people would start investing on their lives – this has been another motive for this research.

* *

This research has made the author rethink about the meaning of the word *peer-to-peer*.

As a recommended direction for future work, this dissertation proposed use of WAT/*i*-WAT to support recovery of communities after catastrophic events. It is a proposal of a model in which everyone in the world can help each other as *peers*.

Our world seems to be undergoing difficulties, many of which are beyond us tiny existences to handle. But still our lives must go on. The author is in the hope that those people affected by disasters will consider WAT/*i*-WAT as an option for helping themselves.

When disasters hit us, we would feel as if we lost everything. However, as long as we are alive, we have a chance for the future, because our skills and experiences should help ourselves. Even if we lack funds, we can create relationships of mutual help, paying back for their time by our time.

The author believes that presence of the Internet is essential for realizing such collaboration in the planetary scale. Hopefully, this research has contributed, and will continue to contribute to make this wonderful environment even a better place.

Appendix A

Protocols

A.1 Public Key Exchange Protocol

This section explains the protocol by an example in which Alice requests Bob for his public key, to which Bob replies by sending his key.

A.1.1 Request

Example 1: Alice requests Bob for his public key

```
<message to="bob@media-art-online.org">
  <body/>
  <x xmlns="http://www.media-art-online.org/gnupg/">
    <get jid="bob@media-art-online.org"/>
  </x>
</message>
```

jid attributed in `<get/>` element specifies the Jabber ID to which the public key in need is bound. The requester can ask for any public keys the peer binds to a Jabber ID, not necessarily to their own. This way, a requester can ask for their own public keys signed by the peer in order to update their keys and have a better chance that the key is validated by their prospective trade partners.

A.1.2 Public Key Transfer

Example 2: Bob sends to Alice his public key

```
<message from="bob@media-art-online.org/wijabot"
  to="alice@media-art-online.org/wija" xmlns="jabber:client">
  <body/>
  <x xmlns="http://www.media-art-online.org/gnupg/">
    <put jid="bob@media-art-online.org">
      mQGibEHYEM4RBADAJPOyInKdf6Be4UfSU5CofrjRePRsrf8U4oQUHJvEhE3LSh1jsneHbPRJGMky
      Ymr+FU1j1DP8mewRiswUoGby18AF9dic77X13pkfDkzWjT4mLW+kn35/cAgj+dMAw6A155RfBV+8K
      yts3APXNzKzC77ed9kU1fY57YFVoHBcwCg/OfXmapIkZehMFDzBqDGN6AYDxED/R7EGpmDt13mh/w
      VGNgGxToa8XR81NrLyySgU/rXyCMoYthq3ZyY2Tjck8rbdICnwiGojiZ9J9g3HQ6gMj6/NoMjTLYU
      /19HexHhE/o6MW/F60ZY2aSn3ZuP1EBPR0gvJebuWXDbMbwQk6hfgbupiNbHLDpSwsHgHFkNL/oxf
```

```

mRcA/OeD7bcJnfw7qWKSZEM49tAm9g/Yaba0DUUxNcT0a2GXJsfdD3jQUBcesqBfN3D2I7owNyat+
rgKhHknS5qVY0ceM9mCXisUm8CkjAhs122RQDk3K0yRni/xQ+0vYcTDSKmrFyT8hSDFouzeQ2kx+N
8PfQNMgwjIeRetaefCRD3ArQicHJveHkgPHByb3h5QG11ZG1hLWFydC1vbmXPbmUub3JnPohEBBMR
AgAeBQJB2BD0AhsDBgsJCAcDAgMVAgMDFgIBAh4BAheAAAoJEPXdXacWAQ7mstUAn23ysN1zJIC69
d3JZi2E/+ppf9V5AJ95yWdAwNvBAfHEg7jrka1zaZe147kBDQRB2BE4EAQA164dDb8TvGZvSSafn1
Acs/mrnhT3q7E9fyTd747iCYN80Zm5R05CHV0KjTTvibovJQp8As273tJkmewQ5uWm1yZFDA31BJ6
vstkIripVcxtplc+p8bvncNtCV1HrwZJOZVNI/MF8EmpcJsRa6UdxrjLGH23wdIes1lukNz13Qn8A
AwUD/iPjZn7G1f0ScUCFFzei9a72ec1jJ4ZwRnLIpJ2CCF80nWuNXmByDQUyZvZ72K7L3Ekzj8Ri4
kAAwEikfeZFQhLvL1AqysxSIHz8W/FtBACHv8KJRj9qBYeDKnD6dVfr/ObxqR3uuC4LjDhnQOBQSZ
7ooY+Q00rWUnJjIfwYAqJXiEkEGBECAAKfAkHYETgCGwwACgkQ9d1dpxYBDuac+ACg3mcs2TBEliF
qRhZ507033FMS8KAAnj5/nZOANBcrYcebeedPcCXsSl/7
</put>
</x>
</message>

```

The client is advised to present the calculated fingerprint to the user *before* binding the public key with the Jabber ID, or allow the user to *undo* binding if the authenticity of the public key was found dubious.

A.2 *i*-WAT Protocol

This section explains the protocol by an example in which Alice issues a vanishing *reduction* ticket in barter dollar to Bob, Bob uses it to Cameron, and Cameron uses it to Alice for a redemption.

A.2.1 Issuing

Example 1: Alice draws a *reduction* ticket

```

<message to="bob@media-art-online.org">
<body/>
<x seq="110" xmlns="http://www.media-art-online.org/iwat/">
<item>Thanks for the lift!</item>
<signed>
<draw creditor="bob@media-art-online.org"
debtor="alice@media-art-online.org" id="1462394433">
<sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
<min>0</min>
<var per="week">
<constant value="-0.0010"></constant>
</var>
<memo>I will draw a dog for you.</memo>
</draw>
</signed>
<signature>
iD8DBQBDuHu9dz1H60eon3cRantXAJ49dSG41hEjcxqLfhTUtVMc2xxSuACeI5I7HKiyLXswsa
HHcCLv4DKAONQ=
</signature>
</x>
</message>

```

Example 2: Bob acknowledges the receipt of the message

```

<message from="bob@media-art-online.org/wonderland"
to="alice@media-art-online.org/wija" xmlns="jabber:client">
<body/>

```



```
<x ack="110" xmlns="http://www.media-art-online.org/iwat/" />
</message>
```

Example 3: Bob accepts the ticket

```
<message from="bob@media-art-online.org/wonderland"
to="alice@media-art-online.org" xmlns="jabber:client">
<body/>
<x seq="46" xmlns="http://www.media-art-online.org/iwat/">
<item>Re: Thanks for the lift!</item>
<signed>
<accept>
<signed>
<draw creditor="bob@media-art-online.org"
debtor="alice@media-art-online.org" id="1462394433">
<sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
<min>0</min>
<var per="week">
<constant value="-0.0010"></constant>
</var>
<memo>I will draw a dog for you.</memo>
</draw>
</signed>
<signature>
iD8DBQBdUHu9dz1H60eon3cRAntXAJ49dSG4lhEjcXqLfhtUtVMc2xxSuACeI5I7HKiyLXswsa
HHcCLv4DKAONQ=
</signature>
</accept>
</signed>
<signature>
iD8DBQBdUHuHxMqqnJEHJZCFgRAjXhAJ953/712XrloWQvj8+HaYa1zYv1EACeL3dZQdAjhONGQIb
bnovw8Jcal6I=
</signature>
</x>
</message>
```

Example 4: Alice acknowledges the receipt of the message

```
<message to="bob@media-art-online.org/wonderland">
<body/>
<x ack="46" xmlns="http://www.media-art-online.org/iwat/" />
</message>
```

Example 5: Alice approves the transaction

```
<message to="bob@media-art-online.org">
<body/>
<x seq="111" xmlns="http://www.media-art-online.org/iwat/">
<signed>
<approve>
<signed>
<draw creditor="bob@media-art-online.org"
debtor="alice@media-art-online.org" id="1462394433">
<sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
<min>0</min>
<var per="week">
<constant value="-0.0010"></constant>
</var>
<memo>I will draw a dog for you.</memo>
</draw>
```

```

    </signed>
    <signature>
      iD8DBQBDuHu9dz1H60eon3cRAntXAJ49dSG4lhEjcXqLfhtUtVMc2xxSuACeI5I7HKiyLXsww
      aHhcCLv4DKA0NQ=
    </signature>
  </approve>
</signed>
<signature>
  iD8DBQBDuHysdz1H60eon3cRArnpAJ9ASPIRjK41lt5akorQfDvtiH3PVgCeODGHoDmGgQra/C/
  Myx2WvXmI+kk=
</signature>
</x>
</message>

```

Example 6: Bob acknowledges the receipt of the message

```

<message from="bob@media-art-online.org/wonderland"
  to="alice@media-art-online.org/wija" xmlns="jabber:client">
  <body/>
  <x ack="111" xmlns="http://www.media-art-online.org/iwat/" />
</message>

```

A.2.2 Circulation

Example 7: Bob uses the ticket to Cameron

```

<message to="cameron@media-art-online.org">
  <body/>
  <x seq="47" xmlns="http://www.media-art-online.org/iwat/">
    <item>Thanks for the book!</item>
    <signed>
      <use creditor="cameron@media-art-online.org"
        user="bob@media-art-online.org">
        <signed>
          <draw creditor="bob@media-art-online.org"
            debtor="alice@media-art-online.org" id="1462394433">
            <sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
            <min>0</min>
            <var per="week">
              <constant value="-0.0010"></constant>
            </var>
            <memo>I will draw a dog for you.</memo>
          </draw>
        </signed>
        <signature>
          iD8DBQBDuHu9dz1H60eon3cRAntXAJ49dSG4lhEjcXqLfhtUtVMc2xxSuACeI5I7HKiyLXsww
          aHhcCLv4DKA0NQ=
        </signature>
      </use>
    </signed>
    <signature>
      iD8DBQBDuHzzxqjnJEHJZCFgRAgJ+AKDFK7Q260dfTE/vDd06j6i69SW02wCbBwfEXTqzHxy9gth
      rBDxzagmjXGI=
    </signature>
  </x>
</message>

```

Example 8: Cameron acknowledges the receipt of the message

```

<message from="cameron@media-art-online.org/wija"

```

```

to="bob@media-art-online.org/wonderland" xmlns="jabber:client">
<body/>
<x ack="47" xmlns="http://www.media-art-online.org/iwat/">
</message>

```

Example 9: Cameron accepts the ticket and tells Bob

```

<message from="cameron@media-art-online.org/wija"
to="bob@media-art-online.org" xmlns="jabber:client">
<body/>
<x seq="70" xmlns="http://www.media-art-online.org/iwat/">
<item>Re: Thanks for the book!</item>
<signed>
<accept>
<signed>
<use creditor="cameron@media-art-online.org"
user="bob@media-art-online.org">
<signed>
<draw creditor="bob@media-art-online.org"
debtor="alice@media-art-online.org" id="1462394433">
<sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
<min>0</min>
<var per="week">
<constant value="-0.0010"></constant>
</var>
<memo>I will draw a dog for you.</memo>
</draw>
</signed>
</signed>
<signature>
iD8DBQBDuHu9dz1H60eon3cRAntXAJ49dSG41hEjcxQlFhTUtVMc2xxSuACeI5I7HKiyLXsw
saHHcCLv4DKA0NQ=
</signature>
</use>
</signed>
<signature>
iD8DBQBDuHxzqqnJEHJZCFgRAgJ+AKDFK7Q260DfTE/vDd06j6i69SW02wCbBwfEXTqzHxy9g
thrBDxzagmjXGI=
</signature>
</accept>
</signed>
<signature>
iD8DBQBDuHOCY4odyrFtFsURAsH4AJ4y+o7vDIY1PDVu9AJARbbCO+vSIgCfXdg8Dwm6IXwWnr/
t/uPOLxptFPY=
</signature>
</x>
</message>

```

Example 10: Bob acknowledges the receipt of the message

```

<message to="cameron@media-art-online.org/wija">
<body/>
<x ack="70" xmlns="http://www.media-art-online.org/iwat/">
</message>

```

Example 11: Cameron queries Alice about the transaction

```

<message from="cameron@media-art-online.org/wija"
to="alice@media-art-online.org" xmlns="jabber:client">
<body/>
<x seq="71" xmlns="http://www.media-art-online.org/iwat/">

```

```

<signed>
  <accept>
    <signed>
      <use creditor="cameron@media-art-online.org"
        user="bob@media-art-online.org">
        <signed>
          <draw creditor="bob@media-art-online.org"
            debtor="alice@media-art-online.org" id="1462394433">
            <sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
            <min>0</min>
            <var per="week">
              <constant value="-0.0010"></constant>
            </var>
            <memo>I will draw a dog for you.</memo>
          </draw>
        </signed>
      <signature>
        iD8DBQBDuHu9dz1H60eon3cRAnTxAJ49dSG41hEjcXqLfhtUtVMc2xxSuACeI5I7HKiyLXs
        wsAHcCLv4DKAONQ=
      </signature>
    </use>
  </signed>
  <signature>
    iD8DBQBDuHzzxqgnJEHJZCFgRAgJ+AKDFK7Q260DfTE/vDd06j6i69SW02wCbBwfEXTqzHxy9g
    thrBDxzagmjXGI=
  </signature>
</accept>
</signed>
<signature>
  iD8DBQBDuHOCY4odyrFtFsURAsH4AJ4y+o7vDIY1PDVu9AJARbbC0+vSIgCfXdxg8Dwm6IXwWnr/
  t/uPOLxptFPY=
</signature>
</x>
</message>

```

Example 12: Alice acknowledges the receipt of the message

```

<message to="cameron@media-art-online.org/wija">
  <body/>
  <x ack="71" xmlns="http://www.media-art-online.org/iwat/">
</message>

```

Example 13: Alice approves the transaction and tells Cameron

```

<message to="cameron@media-art-online.org">
  <body/>
  <x seq="112" xmlns="http://www.media-art-online.org/iwat/">
  <signed>
    <approve>
      <signed>
        <use creditor="cameron@media-art-online.org"
          user="bob@media-art-online.org">
          <signed>
            <draw creditor="bob@media-art-online.org"
              debtor="alice@media-art-online.org" id="1462394433">
              <sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
              <min>0</min>
              <var per="week">
                <constant value="-0.0010"></constant>
              </var>
              <memo>I will draw a dog for you.</memo>
            </draw>
          </signed>
        </use>
      </signed>
    </approve>
  </signed>
</x>
</message>

```

```

    </draw>
  </signed>
  <signature>
    iD8DBQBDuHu9dz1H60eon3cRAntXAJ49dSG41hEjcXqLfhtUuVMc2xxSuACeI5I7HKiyLXs
    wsaHHcCLv4DKAONQ=
  </signature>
</use>
</signed>
<signature>
  iD8DBQBDuHzzxqqnJEHJZCFgRAgJ+AKDFK7Q260DfTE/vDd06j6i69SW02wCbBwfEXTqzHxy9g
  thrBDxzagmjXGI=
</signature>
</approve>
</signed>
<signature>
  iD8DBQBDuH0wdz1H60eon3cRAsjPAJ95P1nY/OE8u+dkYZbMtlJti8FzLQCfYYYcKZRky/Ova7Q
  rUDF0fs7zjg4=
</signature>
</x>
</message>

```

Example 14: Cameron acknowledges the receipt of the message

```

<message from="cameron@media-art-online.org/wija"
  to="alice@media-art-online.org/wija" xmlns="jabber:client">
  <body/>
  <x ack="112" xmlns="http://www.media-art-online.org/iwat/">
</message>

```

Example 15: Alice also tells Bob about the approval

```

<message to="bob@media-art-online.org">
  <body/>
  <x seq="113" xmlns="http://www.media-art-online.org/iwat/">
    <signed>
      <approve>
        <signed>
          <use creditor="cameron@media-art-online.org"
            user="bob@media-art-online.org">
            <signed>
              <draw creditor="bob@media-art-online.org"
                debtor="alice@media-art-online.org" id="1462394433">
                <sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
                <min>0</min>
                <var per="week">
                  <constant value="-0.0010"></constant>
                </var>
                <memo>I will draw a dog for you.</memo>
              </draw>
            </signed>
          <signature>
            iD8DBQBDuHu9dz1H60eon3cRAntXAJ49dSG41hEjcXqLfhtUuVMc2xxSuACeI5I7HKiyLXs
            wsaHHcCLv4DKAONQ=
          </signature>
        </use>
      </signed>
    <signature>
      iD8DBQBDuHzzxqqnJEHJZCFgRAgJ+AKDFK7Q260DfTE/vDd06j6i69SW02wCbBwfEXTqzHxy9g
      thrBDxzagmjXGI=
    </signature>
  </approve>

```

```

</signed>
<signature>
  iD8DBQBDuH0wdz1H60eon3cRAsjPAJ95P1nY/OE8u+dkYZbMt1Jti8FzLQCfYYYcKZRky/Ova7Q
  rUDF0fs7zjg4=
</signature>
</x>
</message>

```

Example 16: Bob acknowledges the receipt of the message

```

<message from="bob@media-art-online.org/wonderland"
  to="alice@media-art-online.org/wija" xmlns="jabber:client">
  <body/>
  <x ack="113" xmlns="http://www.media-art-online.org/iwat/">
</message>

```

A.2.3 Redemption

Example 17: Cameron uses the ticket to Alice

```

<message from="cameron@media-art-online.org/wija" to="alice@media-art-online.org"
  xmlns="jabber:client">
  <body/>
  <x seq="72" xmlns="http://www.media-art-online.org/iwat/">
  <item>Thanks for all the fish!</item>
  <signed>
    <use creditor="alice@media-art-online.org"
      user="cameron@media-art-online.org">
      <signed>
        <use creditor="cameron@media-art-online.org"
          user="bob@media-art-online.org">
          <signed>
            <draw creditor="bob@media-art-online.org"
              debtor="alice@media-art-online.org" id="1462394433">
              <sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
              <min>0</min>
              <var per="week">
                <constant value="-0.0010"></constant>
              </var>
              <memo>I will draw a dog for you.</memo>
            </draw>
          </signed>
        </signed>
      </signed>
    </use>
  </signed>
  <signature>
    iD8DBQBDuHu9dz1H60eon3cRAntXAJ49dSG41hEjcXqLfhtUtVMc2xxSuACeI5I7HKiyLXs
    wsaHHcCLv4DKAONQ=
  </signature>
  </use>
</signed>
<signature>
  iD8DBQBDuHxzqqnJEHJZCFgRAgJ+AKDFK7Q260DfTE/vDd06j6i69SW02wCbBwfEXTqzHxy9g
  thrBDxzagmjXGI=
</signature>
</use>
</signed>
<signature>
  iD8DBQBDuH1wY4odyrFtFsURAKvQAJ9QjF1BojSgl7GnkSIT7JYZtbPzXwCdH2A3gcXM2/T1i15
  4i0BbISbeBFU=
</signature>
</x>
</message>

```

Example 18: Alice acknowledges the receipt of the message

```
<message to="cameron@media-art-online.org/wija">
  <body/>
  <x ack="72" xmlns="http://www.media-art-online.org/iwat/">
</message>
```

Example 19: Alice approves the transaction

```
<message to="cameron@media-art-online.org">
  <body/>
  <x seq="114" xmlns="http://www.media-art-online.org/iwat/">
    <signed>
      <approve>
        <signed>
          <use creditor="alice@media-art-online.org"
            user="cameron@media-art-online.org">
            <signed>
              <use creditor="cameron@media-art-online.org"
                user="bob@media-art-online.org">
                <signed>
                  <draw creditor="bob@media-art-online.org"
                    debtor="alice@media-art-online.org" id="1462394433">
                    <sum ns="http://www.media-art-online.org/dollar/#var">1</sum>
                    <min>0</min>
                    <var per="week">
                      <constant value="-0.0010"></constant>
                    </var>
                    <memo>I will draw a dog for you.</memo>
                  </draw>
                </signed>
              <signature>
                iD8DBQBDuHu9dz1H60eon3cRAntXAJ49dSG4lhEjcXqLfhTUtVMc2xxSuACeI5I7HKiyL
                XwsaHHcCLv4DKAONQ=
              </signature>
            </use>
          </signed>
          <signature>
            iD8DBQBDuHHzxqjnJEHJZCFgRAGJ+AKDFK7Q260dfTE/vDd06j6i69SW02wCbBwfEXTqzHxy
            9gthrBDxzagmjXGI=
          </signature>
        </use>
      </signed>
      <signature>
        iD8DBQBDuH1wY4odyrFtFsURAKvqAJ9QjF1BojSgl7GnkSIT7JYztbPzXwCdH2A3gcXM2/T1i
        154i0BbISbeBFU=
      </signature>
    </approve>
  </signed>
  <signature>
    iD8DBQBDuH2Adz1H60eon3cRALpdAJ9ysGu/eQQBA2YDEONtUpnoTh9BgwCghhsk0UbmVHnNFE2
    7XqHNx0qnR3A=
  </signature>
</x>
</message>
```

Example 20: Cameron acknowledges the receipt of the message

```
<message from="cameron@media-art-online.org/wija"
  to="alice@media-art-online.org/wija" xmlns="jabber:client">
  <body/>
```

```
<x ack="114" xmlns="http://www.media-art-online.org/iwat/" />
</message>
```

A.3 Hypertext Sharing Protocol

This section explains the protocol by an example in which Alice requests Bob for his bookmarks, in which she finds a greeting card, and downloads the image from the Bob's client. It is assumed that Alice always uses a proxy for SOCKS5 bytestreams.

A.3.1 Bookmark Transfer

Example 1: Alice asks Bob for his bookmarks

```
<message to="bob@media-art-online.org/wija">
  <body/>
  <x xmlns="http://www.media-art-online.org/bookmark/">
    <get/>
  </x>
</message>
```

Example 2: Bob returns his bookmarks

```
<message from="bob@media-art-online.org/wija"
  to="alice@media-art-online.org/wija" xmlns="jabber:client">
  <body/>
  <x xmlns="http://www.media-art-online.org/bookmark/">
    <put>
      <link href="xmpp://bob@media-art-online.org/wija/bookmark/postcard.jpg">
        <content>Greeting card 2006</content>
        <description/>
        <file>/Users/bob/Public/postcard.jpg</file>
      </link>
    </put>
  </x>
</message>
```

A.3.2 Hypertext Transfer

Example 3: Alice initiates hypertext streaming with Bob

```
<iq id="4b43d2b6" to="bob@media-art-online.org/wija" type="set">
  <si id="s4b43d2b6" mime-type="application/octet-stream"
  profile="http://www.media-art-online.org/hyper-text-transfer/"
  xmlns="http://jabber.org/protocol/si">
    <hyper-text sub-resource="/bookmark/postcard.jpg"
    xmlns="http://www.media-art-online.org/hyper-text-transfer/" />
    <feature xmlns="http://jabber.org/protocol/feature-neg">
      <x type="form" xmlns="jabber:x:data">
        <field type="single" var="stream-method">
          <option>
            <value>http://jabber.org/protocol/bytestreams</value>
          </option>
          <option>
            <value>http://jabber.org/protocol/ibb</value>
          </option>
        </field>
      </x>
    </feature>
  </si>
</iq>
```



```

    </field>
  </x>
</feature>
</si>
</iq>

```

Example 4: Bob accepts to send the file by a SOCKS5 bytestream

```

10:30:34:
<iq from="bob@media-art-online.org/wija" id="4b43d2b6"
to="alice@media-art-online.org/wija" type="result" xmlns="jabber:client">
<si mime-type="image/jpeg" xmlns="http://jabber.org/protocol/si">
  <file name="06nenga.jpg" size="107013"
xmlns="http://jabber.org/protocol/si/profile/file-transfer"/>
  <feature xmlns="http://jabber.org/protocol/feature-neg">
    <x type="submit" xmlns="jabber:x:data">
      <field var="stream-method">
        <value>http://jabber.org/protocol/bytestreams</value>
      </field>
    </x>
  </feature>
</si>
</iq>

```

Example 5: Alice tells Bob about the proxy

```

<iq id="4b43d2b6" to="bob@media-art-online.org/wija" type="set">
  <query mode="tcp" sid="s4b43d2b6"
xmlns="http://jabber.org/protocol/bytestreams">
    <streamhost host="203.178.143.21"
jid="proxy@media-art-online.org/wijabot" port="8880"/>
  </query>
</iq>

```

Alice has obtained the IP address and the port number from the proxy.

Example 6: Bob accepts to communicate with the proxy

```

<iq from="bob@media-art-online.org/wija" id="4b43d2b6"
to="alice@media-art-online.org/wija" type="result" xmlns="jabber:client">
<query xmlns="http://jabber.org/protocol/bytestreams">
  <streamhost-used jid="proxy@media-art-online.org/wijabot"/>
</query>
</iq>

```

Example 7: Alice activates communication with Bob via the proxy

```

<iq id="4b43d2b6" to="proxy@media-art-online.org/wijabot" type="set">
  <query sid="s4b43d2b6" xmlns="http://jabber.org/protocol/bytestreams">
    <activate>bob@media-art-online.org/wija</activate>
  </query>
</iq>

```

Example 8: The proxy acknowledges the success

```

<iq from="proxy@media-art-online.org/wijabot" id="4b43d2b6"
to="alice@media-art-online.org/wija" type="result" xmlns="jabber:client"/>

```


Appendix B

Simulator

B.1 Acquisition

- The source code of the simulator is available from the following directory in the Perforce[89] depot as described in the *wija* development environment Wiki page[102].

```
//depot/scm/main/java/org/media_art_online/iwatsim/...
```

B.2 Data Description (XML)

The simulator takes data described in XML from a file specified in the command line (default file name is “iwatsim.xml”). If the file name is omitted, and the file does not exist, the simulator creates a new file with the default set of data. Perhaps this is the easiest way to create a template.

The following is the list of major elements:

<min-links/> The recommended minimum number of links in the initial network.

<population/> The number of participants.

<resources/> The number of resources (materials).

<rounds/> The number of rounds.

<rounds-output/> On every what rounds the (partial) results will be output.

<random-seed/> The seed for the pseudo-random number generation.

<type/> Participant type (specifiable more than once).

<class-name/> The full class name for the participant type.

- <**suffix**/> The suffix to be used for the names of output files.
- <**percentage**/> Ratio of participants of this type to the whole population (in percentage; the values must sum to 100).
- <**max-trades-per-round**/> The maximum number of active trades per round.
- <**rate-new-partner**/> The probability to search for a partner up to two-hop away.
- <**rate-default**/> The bankruptcy rate.
- <**interest-credit**/> The interest rate per round to the positive balance of the MCS account.
- <**interest-debit**/> The interest rate per round to the negative balance of the MCS account.
- <**max-debit**/> The maximum negative part of the balance.
- <**rate-consumption**/> The consumption rate per round of the materials which participants of this type produce.
- <**rate-production**/> The production rate per round of the materials which participants of this type produce.
- <**max-mot-ratio**/> The ratio of the maximum value of a *multiplication* ticket to its initial value which participants of this type issue.
- <**min-rot-ratio**/> The ratio of the minimum value of a *reduction* ticket to its initial value which participants of this type issue.
- <**rate-variance**/> The over-time rate per round.
- <**rate-mutation**/> The probability to shift its type to a more advantageous one.

Figure B.1 is an example of the content of a data description file.

B.3 Archetypes

B.3.1 DefaultParticipant (i-WAT user)

Usage of currency: Uses *i*-WAT tickets only.

Condition of active trades: The ownership of the partner's self-product is equal to or more than 1.0, and one's ownership of the resource is less than 1.0.

Searching partners: Randomly chooses a direct acquaintance, and by the probability to extend the search, randomly chooses the direct acquaintance of the chosen participant. The search fails if the chosen participant is themselves.

```

<?xml version="1.0" encoding="UTF-8"?>
<iwatsimp>
  <output-chains>chain-length</output-chains>
  <output-credit-distribution>credit-distribution</output-credit-distribution>
  <output-debit-total>total-debit</output-debit-total>
  <output-credit-welfare>credit-welfare</output-credit-welfare>
  <output-link-distribution>link-distribution</output-link-distribution>
  <output-link-distribution-initial>initial-link-distribution</output-link-distribution-initial>
  <output-network>network</output-network>
  <output-network-initial>initial-network</output-network-initial>
  <output-trades-accumulated>accumulated-number-of-trades</output-trades-accumulated>
  <output-trade-distribution-active>active-trade-distribution</output-trade-distribution-active>
  <output-trade-distribution-passive>passive-trade-distribution</output-trade-distribution-passive>
  <output-welfare-distribution>welfare-distribution</output-welfare-distribution>
  <min-links>3</min-links>
  <population>250</population>
  <resources>100</resources>
  <rounds>500</rounds>
  <rounds-output>100</rounds-output>
  <random-seed>31415926535897932</random-seed>
  <type>
    <class-name>org.media_art_online.iwatsim.BankingParticipant</class-name>
    <suffix></suffix>
    <percentage>100</percentage>
    <max-trades-per-round>3</max-trades-per-round>
    <rate-new-partner>0.2</rate-new-partner>
    <rate-default>0.02</rate-default>
    <interest-credit>0.01</interest-credit>
    <interest-debit>0.01</interest-debit>
    <max-debit>10.0</max-debit>
    <rate-consumption>0.1</rate-consumption>
    <rate-production>3.0</rate-production>
    <max-mot-ratio>2.0</max-mot-ratio>
    <min-rot-ratio>0.0</min-rot-ratio>
    <rate-variance>0.0</rate-variance>
    <rate-mutation>0.0</rate-mutation>
  </type>
</iwatsimp>

```

Figure B.1: An example of the content of a data description file

Condition of drawing: Can draw a new ticket if the debt does not reach the maximum.

How to draw *multiplication* tickets: Draws multiple tickets with the same value which is the required value divided by the ratio of the maximum value. This prevents the situation in which the current value of one ticket exceeds 1.0 (if this happens the ticket is no longer usable in the simulated world).

Selection of using tickets: Selects the tickets first-in, first-out order from the list of acquired tickets.

Addition of links to/from the partners: If there is not a link yet, a link is added.

Addition of links to/from the drawers: If the receiver of a ticket does not have a link to the drawer, a new link is added.

B.3.2 RedeemingParticipant

Priority on redemption: When choosing an *i*-WAT ticket to use, choose the one causes a redemption if there is such one.

This type behaves as DefaultParticipant otherwise.

This type is created so that the effectiveness of EV1 can be independently measured.

B.3.3 StretchingParticipant

Priority on tickets with longer chains: When choosing an *i*-WAT ticket to use, choose the one whose chain is longer than others unless the type of the partner is NoStretchingParticipant.

This type behaves as RedeemingParticipant otherwise.

This type is created so that the effectiveness of EV2 can be independently measured.

B.3.4 SelectiveParticipant

Priority on drawers: When choosing a partner, by 10% chance, randomly select a partner from the drawers of acquired tickets, and if the selected one is ready to trade, choose that participant.

This type behaves as StretchingParticipant otherwise.

This type is created so that the effectiveness of EV3 can be independently measured. This type also represents the case where all standard evasive actions are incorporated.

B.3.5 OriginalWATParticipant

Usage of currency: uses WAT tickets only.

No addition of links to drawers: When using a WAT ticket, receivers are not added a new link.

This type behaves as SelectiveParticipant otherwise.

B.3.6 BankingParticipant

Usage of currency: Uses the MCS account only.

Loan: If the balance does not reach the limit of debt, get a loan from the MCS.

Adjustment to i -WAT optimization: When choosing a partner, by 10% chance, randomly selects one from the direct acquaintances, and if the one is ready to trade, choose that one.

This type behaves as DefaultParticipant otherwise.

B.3.7 GlobalMarketBankingParticipant

Choosing partners: Randomly chooses from the full population. The search fails if the chosen one is themselves.

No addition of links to the partners: The network is not altered by trades.

Adjustment to i -WAT optimization: When choosing a partner, by 10% chance, randomly selects one from the direct acquaintances, and if the one is ready to trade, choose that one.

This type behaves as BankingParticipant otherwise.

B.3.8 BankingWATParticipant

Choosing currency: If the partner is a BankingParticipant, uses the MCS, but never makes their own balance of the MCS account negative.

This type behaves as BankingParticipant and OriginalWATParticipant otherwise.

B.3.9 PlaceboParticipant

Random linking: Instead of adding a link to/from the drawer, adds a new link to/from a randomly chosen participant.

This type behaves as DefaultParticipant otherwise.

This type is created so that the effectiveness of having a link to the drawers can be measured.

B.3.10 PlaceboRedeemingParticipant

Random linking: Instead of adding a link to/from the drawer, adds a new link to/from a randomly chosen participant.

This type behaves as RedeemingParticipant otherwise.

This type is created so that the effectiveness of having a link to the drawers can be measured.

B.3.11 PlaceboStretchingParticipant

Random linking: Instead of adding a link to/from the drawer, adds a new link to/from a randomly chosen participant.

This type behaves as StretchingParticipant otherwise.

This type is created so that the effectiveness of having a link to the drawers can be measured.

B.3.12 PlaceboSelectiveParticipant

Random linking: Instead of adding a link to/from the drawer, adds a new link to/from a randomly chosen participant.

This type behaves as SelectiveParticipant otherwise.

This type is created so that the effectiveness of having a link to the drawers can be measured.

B.3.13 SemiOptimizedParticipant

Optimization to *reduction* tickets: Tries to choose tickets whose expected loss by holding is greater than others.

Optimization to *multiplication* tickets: Not to choose the tickets until the values reach the maximum.

This type behaves as SelectiveParticipant otherwise.

B.3.14 OptimizedParticipant

Optimization to *reduction* tickets: Not to choose redemption until the value reaches the minimum.

Optimization to *multiplication* tickets: Tries to choose the tickets for redemption.

This type behaves as SemiOptimizedParticipant otherwise.

B.4 Command

The simulator is operated by the following command lines.

```
$ java -Xmx300M -jar iwatsim.jar [FILE]
$ sort chain-length | uniq -c | sort -r > chain-length-distribution
$ sort chain-length-x | uniq -c | sort -r > chain-length-distribution-x
```

Depending on the settings, the Java virtual machine may require more memory than 300MB.

B.5 Output Files and Formatting

(Partial) results are written to the following files, which can be formatted into graphs by using some software tools, where

- *-n* denotes a round,
- *-x* is the suffix for the participant type (used only when there are more than one participant type in the simulation),
- and the software tool to be used is *gnuplot*[110] unless otherwise specified.

chain-length-distribution-*x* The distribution of the chain lengths of tickets.

```
> plot "chain-length-distribution-x" using 2:1
```

or, by using *R*[96],

```
> cl <- read.table("chain-length-x", header=FALSE)
> hist(cl$V1)
```

initial-link-distribution, **link-distribution- n** (initial) link distribution.

```
> plot "initial-link-distribution", "link-distribution-n",...
```

network- n .net The network (acquaintance relation).

By using *Pajek*[54],

Net → Paths between 2 vertices → Distribution of Distances → From All Vertices

Draw → Draw

The output files tend to be large, and it takes time for the simulator to produce the data. This calculation can be suppressed by omitting the elements `<output-network/>` and `<output-network-initial/>` from the data description file.

welfare-distribution- $x-n$ The distribution of welfare.

```
> plot "welfare-distribution-x-n"
```

or, by using *R*,

```
> w <- read.table("welfare-distribution-x-n", header=FALSE)
> boxplot(w$V1)
```

credit-distribution- $x-n$ The distribution of balances.

```
> plot "credit-distribution-x-n"
```

credit-welfare- $x-n$ The distribution of balances, welfare, initial-links and links

```
> plot "credit-welfare-x-n" using a:b
```

The following is the list of sample $a : b$:

2:1 welfare:balance

3:1 initial-links:balance

4:1 links:balance

2:3 welfare:initial-links

2:4 welfare:links

total-debit- n The total debt in the world.

```
> plot "total-debit-n"
```

{active|passive}-trade-distribution-x-n The distribution of the number of {active|passive} trades.

```
> plot "{active|passive}-trade-distribution-x-n"
```

accumulated-number-of-trades-n The accumulated number of trades.

```
> plot "accumulated-number-of-trades-n"
```


Appendix C

Descriptive Replies to Questionnaires

C.1 WIDE Hours

Replies to questionnaire at the Spring 2004 experiment.

C.1.1 Replies to Question 14 (Impression)

Q14. What is your impression about WIDE Hours?

- I have installed gpg, but I was hesitant to install Java. It would be stressful to run Java on a machine with limited main memory. . .
- It was not clear what the objectives of this experiment was, and what are the expected outcomes.
- I did not realize the experiment was ongoing because I was late. I am sorry.
- I do not seem to understand it fully, sorry.
- I did not try it.
- I did not feel like using it.
- I do not seem to understand the heart of it, what it is for.
- It is interesting. I do not have a concrete idea how this is going to be helpful to us, but there may be various ways to deploy it, and I am looking forward to seeing it. What is WIDE Mirror, by the way?
- I was surprised to receive WIDE Hours after making a comment on the panel.

- What I discovered through this is that oral presentation is not suitable for explaining a complicated concept. A poster presentation would have been better.
- I like it very much.
- I would have liked it if it had completeness (reliability) as the real currencies.
- I could understand WIDE Hours, but I do not understand how WIDE Power is calculated. More explanations would have helped.
- I did not participate. I was absent the first day, so I did not know what was going on.
- I had an impression that oh, you were still doing it. I did not realize that you were.
- I did not use it.
- I do not understand it well.
- It is difficult to have a standard on how I distribute my hours.
- Organize well.
- I do not know if it is going to be useful. To know, we just have to experiment.
- I was not aware of it.
- I think we can use it for something. I would like to use it in my office and at home. Are the specifications and the code open?
- I did not think that the usefulness of this was well presented.
- The objective of WIDE Hours is not clear. I have an impression that it is only for collecting points.

C.1.2 Replies to Question 17 (Interfaces)

Q17. Please tell us how human interfaces of WIDE Hours Web or *wija* can be improved.

- Due to the performance of my machine, Java was quite heavy, and I could not use it as a communication tool. I was forced to wait for 10 seconds to do anything. . .
- I did not understand the intention of it.

- I will allocate my time to understand it. I am sorry.
- I am sorry. I could not join because I did not know the server name.
- I think it failed to motivate people because the value and the meaning of it were difficult to understand.
- I suggest to make a WIDE Hour dispenser or checker by combining it with ID tags.
- I regret that I did not use it well.
- I cannot tell who is whom by the WIDE numbers only.
- I suggest that the all accounts start from zero during the WIDE camp period, so that everyone can be more competitive.
- I would like it to be usable unconsciously. Examples: chairs give hours to the authors of submitted memos, people can send hours to others by putting their name cards over their PCs, etc.
- I am afraid there was not much of an appeal. You could have used the lunch time and so on for announcements.
- I would like to have a mechanism to exchange hours using tags instead of PCs.
- I did not have time to use *wija*. Suggestions of actual usage scenes would have helped.
- I would like it if hours had a link to the WIDE database.
- It does not look cute.
- I do not think there is a manual. I would like more thoughtful descriptions on how to use it, how to collect points.
- It is bothersome to install it.

C.1.3 Replies to Question 20 (Usage)

Q20. What do you think are possible usages of WIDE Hours?

- Some daily enjoyments such as sweets or drinks as rewards during the wine times?
- Resolution.
- I have been using it for showing my appreciations to the loggers of BoFs.

- As a part of WIDE Awards.
- For management of labors among officials at national universities. It seems as if their principle is to get as less work as possible, and this sort of things will be useful as an incentive mechanism.
- I do not seem to have any ideas still.
- For expressing our gratitudes to the PCs.
- As payments to sweets.
- For reserving bandwidth, getting better grade of drinks.
- As incentives to making comments.
- How about using it for setting priorities to reserving BoF time slots?
- As daily expressions of gratitudes.
- I think it is a tool which enables us to appreciate those people doing their best in something in a visible way.
- I would like it if I can drink coffee using WIDE Hours.
- For appreciation in open source communities.

C.1.4 Replies to Question 21 (Improvements)

Q21. How do you think WIDE Hours can be improved?

- *wija* seems to be less usable than the web + RFID version we saw last time. It may be interesting if there is a service to provide better bandwidth and latencies to people with higher WIDE Powers.
- I would like repetitive tutorials, not just once, because it seems that if the first chance is missed, there is no other chance to join.
- Perhaps pre-defined options to choose from may help when we describe reasons to send hours. For example, a mechanism to send a certain amount of hours to a certain task such as issuing a WIDE member certificate, I think, may be useful.
- In three words, hard to understand.
- Resistance to misbehaviors.
- I think that you should put priorities to research discussions on modeling contributions and getting consensus by active research presentations, instead of building applications.

- It is not clear whether this is about evaluating people or making currencies.
- I would like it to be more easily usable.
- I would like something like an orkut-wrapper so that acquaintance relations can be utilized through setting some cost for introducing people to others.
- I think there needs to be more sociological discussions.
- I would like some incentives that motivate us to use it, because what WIDE Hours can realize is not very clear at the moment.
- I am afraid that mixture of many local currencies will be as confusing as having many point cards for different shops.
- I would like more applications to use it.
- I could find no one in Jabber.

C.2 Vegetable Trading

C.2.1 Replies to Question 4 (Usefulness)

Q4. Do you think complementary currencies will be useful to your life? (and why?)

- Reasons for YES
 - I am not ready to feel the possibilities, but I guess it might work in some specific places (age 35, green pepper).
 - I think it can protect us from inflation (age 29, carrot).
 - I may be able to buy things abroad (age 23, eggplant).
 - It's convenient. If this is actualized, it would be safer (age 23, eggplant).
 - I cannot trust the system yet, but I guess there are some situations which would require things like this (age 24, onion).
 - It's convenient (age 19, onion).
 - It will become convenient (age 50, onion).
 - It would be useful in emergencies (age 37, potato).
- Reasons for NO
 - It does not seem easy to use (age 25, carrot).

- Legal tenders are enough (age unknown, potato).
 - Bartering through human-to-human communication will survive until the very end. I wonder if it is good to depend on electronic devices (age 29, potato).
 - I think it wouldn't be useful in real life as long as we see it as a form of currency (age unknown, green pepper).
 - Trust is not guaranteed (age 25, eggplant).
 - If a region is full of various products, then I think that complementary currencies will function well there as the expression of common values among people. If not, direct bartering seems to be sufficient (age 25, carrot).
 - I do not seem to understand complementary currencies well (age 25, office).
 - It does not feel familiar (age 27, office).
 - I have a difficulty familiarizing myself with the idea (age 29, onion).
 - I do not think it can be of immediate use, but I do anticipate for the future. Yet, it might not be for everyone because of too many steps (age 24, green pepper).
 - It is difficult for me to understand how to operate (age 40, potato).
 - I cannot think of a situation in which this will be useful (age 20, office).
- Others
 - I do not seem to understand well (age 52, carrot).
 - ? (age 21, eggplant)
 - I do not seem to understand it fully (age 24, office).
 - It depends on how to use it. NO if it is for trading of vegetables (age 29, green pepper).
 - It would become useful if it is easier to understand how to use it (age 34, green pepper).
 - I could not understand complementary currencies very well (age 22, eggplant).

C.2.2 Replies to Question 5 (Suggestions or Other Thoughts)

Q5. Please tell us any suggestions or other thoughts about *i*-WAT during or after participating in this game.

- I think it is not intuitive to realize that a ticket is acquired. I felt that in order for it to be useful (+usable), a counteraction toward instability of the system is a necessity (age 35, green pepper).
- I suggest improved stability of the system and universal (children and elderlies) access. Also some compatibility among terminals (age 29, carrot).
- I request the terminals to be easier to use. I felt that it would be OK to authenticate just by exchanging messages between the partners, without talking to the servers [author's note: this is a misunderstanding of the authentication system of *i*-WAT, but the author humbly accepts this as it must have looked that way because of the user interface]. That would be faster. It was difficult for me to understand how to use it, especially the screen with vegetable names (age 25, carrot).
- Better communication infrastructure. Improved stability of the operating system. The layers of menus are too deep [author's note: this participant must have used hidden user interface for maintenance]. The simpler is the better. Bigger icons may help, too (age unknown, potato).
- It would help if the tool itself is more stable. I have experienced a number of freezes. The pen is easy to slip, and I could not control it at will. Buttons instead of a pen may be better. The panel is too dark. I could not understand the rule of the game very well (age 29, potato).
- I think the values are different between getting as many vegetables and getting as many points (I do not see a relation that can connect the two). During the game, I saw, a lot of times, that participants were concentrating on collecting as many vegetables, as if that was the whole point of the game. There were only a few who thought it was a game to collect WAT tickets (in fact, I was denied my offer to buy a vegetable with my WAT ticket) (age unknown, green pepper).
- Please stop devices from freezing or keeping to show the black screen [author's note: console for Pocket GnuPG]. Make it easier to exchange information just by bringing two terminals close to each other (age 23, eggplant).
- It was so interesting. But I hope that the machine would be more better at working. Too many problems that made us sad (age 23, eggplant)
- Too many steps for authentication (age 52, carrot).

- I could not understand as the game did not work for me because of a hardware problem. . . It was difficult to operate the device. I did not understand what it was processing. . . (age 21, eggplant).
- Please make the device stop communicating when I do certain things, because it stops me from doing things that I want (age 24, onion).
- Perhaps it would help if a lecture is given on the rules of the game and operation of devices before starting the game, including the lecture on public key cryptography. It would be better just to show “processing” than showing the console window (age 25, eggplant).
- I see vegetables in front of me, but it requires operation of machines to exchange or receive them. It is a waste of time. I didn’t understand the rules. Need more explanations. The screen is too small on the device, and the touch panel is not smooth to operate. If this is supposed to be used at times of post-catastrophe, it should be simpler to use, so that elderlies or children can understand how to use it (age 25, carrot).
- It needs bigger buttons for everyone to be able to use it. Perhaps it would be better if we could use physical buttons on the device instead of the touch panel [author’s note: the software was in fact made operable with the physical buttons, but the author decided not to advertise it to the participants after discussions with testers from a preliminary experiment] (age 25, office).
- Wouldn’t this game require some knowledge on economics to participate? I wish there were more people who could explain things (age 27, office).
- I had nothing to do during the game (age 24, office).
- I found that the game was fun, but it would be better if it were easier to understand (age 34, green pepper).
- I suggest easy, understandable and usable terminals and the system. There will be small children and elderlies among the victims of a disaster. I hope it will be quick and easy to use, just by pressing a button (age 34, green pepper).
- I am anticipating the system to be easier to understand, with GUIs, because this is a local currency system and everyone should be able to use it including elderlies, without requiring too much (age 40, potato).
- It was too complicated for me to understand. I could not understand *i*-WAT or complementary currencies. I have traded vegetables, but I did not know what to do with the ticket and vegetables (age 22, eggplant).

- It was difficult (age unknown, carrot).
- The game is a little difficult to understand. I suggest that a simpler system or explanation will help *i*-WAT to be understood more easily and in depth (age 20, office).
- I think that we need to familiarize ourselves with the idea, so that an increased number of choices which are more practical may help (age 29, onion).
- More explanation is needed to understand it. Perhaps we just need to familiarize ourselves with the idea. Then we can anticipate from it (age 50, onion).
- I suggest more preparation before the game starts (spare batteries and so on). More stepwise explanations would have helped (age 37, potato).
- I did not understand it well. However, it was a lot of fun. I met many people, made new friends (age 19, onion).
- It is questionable whether people would be able to operate devices after some catastrophic events. It would be easier for us to do direct bartering if we could meet each other. The game was fun. But about a half of the participants seemed as if they did not understand the rules. They *were* doing direct bartering. More explanation would have helped. A simpler explanation, such as the following, may have been better: we buy a vegetable with a WAT ticket, and if we got two vegetables, we report. We should clarify the conditions of the devices to be used after catastrophe, such as batteries and ease of use. If we apply *i*-WAT to neighbors, we may find ourselves between contradicting benefits, and I am afraid that the relationship may collapse. This was an educational experience. It was fun (age 29, green pepper).

Bibliography

- [1] Philip E. Agre. P2P and the promise of Internet equality. *Communications of the ACM*, Volume 46(Issue 2), February 2003.
- [2] Apple Computer, Inc. Developer - quicktime. Hypertext document. Available electronically at <http://developer.apple.com/quicktime/>.
- [3] Apple Computer, Inc. Apple - iPod + iTunes, as of 2005. Available electronically at <http://www.apple.com/itunes/>.
- [4] Apple Computer, Inc. Apple - Mac OS X, as of 2005. Available electronically at <http://www.apple.com/macosx/>.
- [5] Auto-ID Center. Auto-ID Center - About The Center. Hypertext document. Available electronically at <http://www.autoidcenter.org/>.
- [6] AZI. Mojo Nation technology overview. Online archive. Available electronically at http://web.archive.org/web/20020127125928/www.mojonation.net/docs/technical_overview.shtml.
- [7] Tim Berners-Lee, Roy T. Fielding, and Henrik Frystyk Nielsen. *Hypertext Transfer Protocol – HTTP/1.0*, May 1996. RFC 1945.
- [8] John Boyer. *Canonical XML Version 1.0*, March 2001. W3C Recommendation. Available electronically at <http://www.w3.org/TR/xml-c14n>.
- [9] Tim Bray, Jean Paoli, C.M.Sperberg-McQueen, and Eve Maler. *Extensible Markup Language (XML) 1.0 (Second Edition)*, October 2000. W3C Recommendation. Available electronically at <http://www.w3.org/TR/REC-xml>.
- [10] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. In *Proceedings of the Royal Society, Volume 426, Number 1871*, 1989.
- [11] Jon Callas, Lutz Donnerhacke, Hal Finney, and Rodney Thayer. *OpenPGP Message Format*, November 1998. RFC 2440.

- [12] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proceedings the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, December 2002.
- [13] ccTsunami.org. Complementary currency tsunami relief center. Hypertext document. Available electronically at <http://www.cctsunami.org/>.
- [14] LETS Chita. Why don't you use CHITA WAT to express your gratitude? Available electronically at <http://lets-chita.circle.ne.jp/chitawat.htm> (*in Japanese*).
- [15] Bram Cohen. Incentives build robustness in bittorrent. In *Proceedings of the First Workshop on Economics of Peer-to-Peer Systems*, May 2003.
- [16] Landon Cox and Brian Noble. Samsara: Honor among thieves in peer-to-peer storage. In *Proceedings of the ACM Symposium on Operating Systems Principles*, October 2003.
- [17] Creative Commons. Creative Commons, as of 2005. Available electronically at <http://creativecommons.org/>.
- [18] Arthur Dahlberg. *When Capital Goes On Strike*. Harper & Brothers Publishers, 1938.
- [19] Stephen E. Deering and Robert M. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*, December 1995. RFC 1883.
- [20] eMercury, Inc. Social networking site [mixi], since 1995. Available electronically at <http://mixi.jp/> (*in Japanese*).
- [21] Joan Feigenbaum and Scott Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM '02)*, September 2002.
- [22] Michal Feldman, Christos Papadimitriou, John Chuang, and Ion Stoica. Free-riding and whitewashing in peer-to-peer systems. In *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 228–236, September 2004.
- [23] Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1*, June 1999. RFC 2616.

- [24] Free Software Foundation, Inc. GNU general public license, 1991. Hypertext document. Available electronically at <http://www.gnu.org/licenses/gpl.html>.
- [25] Free Software Foundation, Inc. The GNU operating system, since 1996. Hypertext document. Available electronically at <http://www.gnu.org/>.
- [26] Ryan Fugger. Money as IOUs in social trust networks & a proposal for a decentralized currency network protocol. Hypertext document. Available electronically at <http://ripple.sourceforge.net/>.
- [27] Sylvio Gesell. *The Natural Economic Order*. The Free Economy Publishing Co., 1934. Translated from the sixth German edition (originally published in 1913). Also available as a hypertext document in English, translated by Phillip Pye, at <http://www.systemfehler.de/en/neo/>.
- [28] Global Public Media. Richard Douthwaite speaks with Julian Darley (Jan 2003). Hypertext document. Available electronically at <http://www.globalpublicmedia.com/interviews/123>.
- [29] Paul Glover. Ithaca HOURS Online. Hypertext document. Available electronically at <http://www.ithacahours.com/>.
- [30] GREE. Home - GREE, since 2004. Available electronically at <http://gree.jp/> (*in Japanese*).
- [31] WIDE Project IDEON Working Group. Ideon-wiki. Hypertext document. Available electronically at <http://member.wide.ad.jp/wg/ideon/pukiwiki.php?en%2FTopPage>.
- [32] WIDE Project IDEON Working Group. wija project. Hypertext document. Available electronically at <http://member.wide.ad.jp/wg/ideon/?en%2FProjects%2Fwija>.
- [33] Vassos Hadzilacos and Sam Toueg. A modular approach to fault-tolerant broadcasts and related problems. Technical Report TR94-1425, Department of Computer Sciecn, Cornell University, May 1994.
- [34] Garrett Hardin. The tragedy of the commons. *Science*, 162, 1968.
- [35] Joe Hildebrand, Peter Millard, Ryan Eatmon, and Peter Saint-Andre. *JEP-0030: Service Discovery*, March 2005.
- [36] Joe Hildebrand and Peter Saint-Andre. *JEP-0115: Entity Capabilities*, October 2004.

- [37] Pak Hyeon-Suk. Han bat lets. Presentation material for Community Currency Summit in EXPO 2005. Available electronically at <http://changeyourmoney.net/> (*in Japanese*).
- [38] IBM Software. WebSphere Everyplace Micro Environment, as of 2005. Available electronically at <http://www-306.ibm.com/software/wireless/weme/>.
- [39] Julie Ingleby. Local economic trading systems: Potentials for new communities of meaning: a brief exploration of eight letsystems, with a focus on decision making. *International Journal of Community Currency Research*, vol.2, 1998. Available electronically at <http://www.geog.le.ac.uk/ijccr/>.
- [40] IRIS project. IRIS: Infrastructure for Resilient Internet Systems. Hypertext document. Available electronically at <http://iris.lcs.mit.edu/>.
- [41] Takaaki Ishida, Shinichi Hisamatsu, Kenji Saito, Masaki Minami, and Jun Murai. Content Cruising System under sparse movements of nodes. In *Proceedings of 2004 Symposium on Applications and the Internet (SAINT 2004 Workshops)*. IEEE Computer Society Press, January 2004.
- [42] Jabber Software Foundation. Jabber: Open instant messaging and a whole lot more, powered by xmpp, since 1999. Available electronically at <http://www.jabber.org/>.
- [43] Justin Karneges. *JEP-0047: In-Band Bytestreams*, December 2003.
- [44] Alexander Komarov. Geek Credit homepage. Hypertext document. Available electronically at <http://home.gna.org/geekcredit/>.
- [45] Marcus Leech. *SOCKS Protocol Version 5*, March 1996. RFC 1928.
- [46] Richard Lethin. Technical and social components of peer-to-peer computing. *Communications of the ACM*, Volume 46(Issue 2), February 2003.
- [47] Linux Online, Inc. The Linux Home Page at Linux Online, as of 2005. Available electronically at <http://www.linux.org/>.
- [48] Maebashi Artists Association. MAAS (Maebashi Artists Association). Hypertext document. Available electronically at <http://www.watsystems.net/users/maassite/>.
- [49] Frederick Mann. Economic means to freedom - part v. Available electronically at <http://www.buildfreedom.com/economic/eco.5.html>.

- [50] Microsoft Corporation. Microsoft Windows Family Home Page, as of 2005. Available electronically at <http://www.microsoft.com/windows/>.
- [51] Peter Millard. *JEP-0020: Feature Negotiation*, May 2004.
- [52] Matthew Miller and Peter Saint-Andre. *JEP-0079: Advanced Message Processing*, November 2005.
- [53] Tim Moreton and Andrew Twigg. Trading in trust, tokens, and stamps. In *Proceedings of the Workshop on the Economics of Peer-to-Peer Systems*, June 2003.
- [54] Andrej Mrvar. Networks / Pajek, as of 2005. Available electronically at <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>.
- [55] Thomas Muldowney. *JEP-0027: Current Jabber OpenPGP Usage*, March 2004.
- [56] Thomas Muldowney, Matthew Miller, and Ryan Eatmon. *JEP-0095: Stream Initiation*, April 2004.
- [57] Thomas Muldowney, Matthew Miller, and Ryan Eatmon. *JEP-0096: File Transfer*, April 2004.
- [58] Jun Murai. *Explorers! of the Wonderful Internet*. TaroJiro-Sha Editus, 2003. (*in Japanese*).
- [59] Murai Lab., Keio University. *ACCIANCO.JP*. <http://www.accianco.jp/>, 2003-2004.
- [60] National Institute of Standards and Technology. *FIPS 180-1 - Secure Hash Standard*, April 1995.
- [61] T.-W. J. Ngan, D. S. Wallach, and P. Druschel. Enforcing fair sharing of peer-to-peer resources. In *2nd International Workshop on Peer-to-Peer Systems (IPTPS)*, Berkeley, California, February 2003.
- [62] Robert Norris and Peter Saint-Andre. *JEP-0086: Error Condition Mappings*, February 2004.
- [63] Jonathan B. Postel. *Internet Protocol*, September 1981. RFC 791.
- [64] Jonathan B. Postel. *Simple Mail Transfer Protocol*, August 1982. RFC 821.
- [65] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communication*, Vol.16(No.4), 1998.

- [66] R.A. Retting, B.N. Persaud, P.E. Garder, and D. Lord. Crash and injury reduction following installation of roundabouts in the united states. *American Journal of Public Health*, Vol.91(No.4), April 2001.
- [67] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, November 2001.
- [68] Peter Saint-Andre. *JEP-0082: Jabber Date and Time Profiles*, May 2003.
- [69] Peter Saint-Andre. *Extensible Messaging and Presence Protocol (XMPP): Core*, October 2004. RFC 3920.
- [70] Peter Saint-Andre. *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence*, November 2004. RFC 3921.
- [71] Peter Saint-Andre. *JEP-0112: User Physical Location*, October 2004.
- [72] Peter Saint-Andre. *JEP-0118: User Tune*, November 2004.
- [73] Peter Saint-Andre. *JEP-0045: Multi-User Chat*, September 2005.
- [74] Kenji Saito. Peer-to-peer money: Free currency over the Internet. In *Proceedings of the Second International Conference on Human.Society@Internet (HSI 2003), Lecture Notes in Computer Science 2713*. Springer-Verlag, June 2003.
- [75] Kenji Saito. Maintaining trust in peer-to-peer barter relationships. In *Proceedings of 2004 Symposium on Applications and the Internet (SAINT 2004 Workshops)*. IEEE Computer Society Press, January 2004.
- [76] Kenji Saito. WOT for WAT: Spinning the web of trust for peer-to-peer barter relationships. *IEICE TRANSACTIONS on Communication*, Vol.E88-B(No.4), April 2005.
- [77] Kenji Saito. Examining the charms of jabber, an extensible instant messaging protocol. *JavaWorld*, January 2006. *in Japanese*.
- [78] Kenji Saito, Eiichi Morino, and Jun Murai. Incentive-compatibility in a distributed autonomous currency system. In *Proceedings of the Fourth International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2005)*, July 2005.

- [79] Kenji Saito, Eiichi Morino, and Jun Murai. Multiplication over time to facilitate peer-to-peer barter relationships. In *Proceedings of the 2nd International Workshop on P2P Data Management, Security and Trust (PDMST '05)*, August 2005.
- [80] Kenji Saito, Eiichi Morino, and Jun Murai. Reduction over time: Easing the burden of peer-to-peer barter relationships to facilitate mutual help. In *Proceedings of the Second International Workshop on Computer Supported Activity Coordination (CSAC 2005)*, May 2005.
- [81] Kenji Saito, Eiichi Morino, and Jun Murai. Reduction over time to facilitate peer-to-peer barter relationships. *IEICE TRANSACTIONS on Information and Systems*, Vol.E89-D(No.1), January 2006.
- [82] Kenji Saito, Eiichi Morino, and Jun Murai. No risk is unsafe: Simulated results on dependability of complementary currencies. In *Proceedings of the First International Conference on Availability, Reliability and Security*, April 2006 (to appear).
- [83] Fred B. Schneider. *On Concurrent Programming*. Springer-Verlag, 1997.
- [84] Jorim Schraven. The economics of community currencies: a theoretical perspective. Unpublished Honours Thesis, Oxford University. Available electronically at <http://www.jorim.nl/>.
- [85] Jorim Schraven. Mutual credit systems and the commons problem: Why community currency systems such as LETS need not collapse under opportunistic behavior. *International Journal of Community Currency Research*, vol.5, 2001. Available electronically at <http://www.geog.le.ac.uk/ijccr/>.
- [86] Fritz Schwarz. Das experiment von Wörgl, 1951. Hypertext document. Available electronically at <http://userpage.fu-berlin.de/~roehrigw/woergl/>, (*Shortened English translation by Hans Eisenkolb is available at <http://www.sunshinecable.com/~eisehan/woergl.htm>*).
- [87] Sidonie Seron. Local Exchange Trading Systems 1 - CREATION AND GROWTH OF LETS. Hypertext document. Available electronically at <http://www.gmlets.u-net.com/resources/sidonie/home.html>.
- [88] Dave Smith, Matthew Miller, and Peter Saint-Andre. *JEP-0065: SOCKS5 Bytestreams*, November 2004.
- [89] Perforce Software. Perforce Software – the fast software configuration management system, 1996, 2005. Hypertext document. Available electronically at <http://www.perforce.com/>.

- [90] Ion Stoica, Robert Morris, M. Frans Kaashoek David Karger, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of ACM SIGCOMM*, August 2001.
- [91] Sun Microsystems, Inc. Java Technology, as of 2005. Available electronically at <http://java.sun.com/>.
- [92] The Eclipse Foundation. Eclipse.org home, as of 2005. Available electronically at <http://www.eclipse.org/>.
- [93] The Free Software Foundation. The GNU Privacy Guard. Hypertext document. Available electronically at <http://www.gnupg.org/>.
- [94] The Free Software Foundation. The GNU Privacy Handbook. Available electronically at <http://www.gnupg.org/>.
- [95] The Internet Engineering Task Force. Ietf home page. Available electronically at <http://www.ietf.org/>.
- [96] The R Project. The R project for statistical computing, as of 2005. Available electronically at <http://www.r-project.org/>.
- [97] Time Dollar USA. Hypertext document. Available electronically at <http://www.timedollar.org/>.
- [98] Vivek Vishnumurthy, Sangeeth Chandrakumar, and Emin Gun Sirer. KARMA: A secure economic framework for p2p resource sharing. In *Proceedings of the Workshop on the Economics of Peer-to-Peer Systems*, June 2003.
- [99] watsystems.net. WATSystems home page. Hypertext document. Available electronically at <http://www.watsystems.net/>.
- [100] WIDE Hour Office. WIDE Hour Web. Hypertext document. Available electronically at <https://fran.sfc.wide.ad.jp/hour/>.
- [101] WIDE Project. WIDE PROJECT Home Page. Hypertext document. Available electronically at <http://www.wide.ad.jp/>.
- [102] WIDE Project IDEON Working Group. *wija development environment*, as of 2005. Available electronically at <http://member.wide.ad.jp/wg/ideon/pukiwiki.php?wija%20development%20environment>.
- [103] Wikipedia. Jeremie Miller - Wikipedia, the free encyclopedia, as of 2005. Available electronically at http://en.wikipedia.org/wiki/Jeremie_Miller.

- [104] Wikipedia. List of Jabber clients - Wikipedia, the free encyclopedia, as of 2005. Available electronically at http://en.wikipedia.org/wiki/List_of_Jabber_clients.
- [105] Wikipedia. Pocket PC - Wikipedia, the free encyclopedia, as of 2005. Available electronically at http://en.wikipedia.org/wiki/Pocket_PC.
- [106] Wikipedia. Replicator (Star Trek) - Wikipedia, the free encyclopedia, as of 2005. Available electronically at http://en.wikipedia.org/wiki/Replicator_%28Star_Trek%29.
- [107] Wikipedia. Spam (electronic) - Wikipedia, the free encyclopedia, as of 2005. Available electronically at <http://en.wikipedia.org/wiki/Spamming>.
- [108] Wikipedia. Spyware - Wikipedia, the free encyclopedia, as of 2005. Available electronically at <http://en.wikipedia.org/wiki/Spyware>.
- [109] Colin C Williams, Theresa Aldridge, Roger Lee, Andrew Leyshon, Nigel Thrift, and Jane Tooke. The role of the third sector in paving a 'third way': Some lessons from local exchange and trading schemes (lets) in the united kingdom. *International Journal of Community Currency Research*, vol.5, 2001. Available electronically at <http://www.geog.le.ac.uk/ijccr/>.
- [110] Thomas Williams and Colin Kelley. gnuplot homepage, as of 2005. Available electronically at <http://www.gnuplot.info/>.
- [111] Beverly Yang and Hector Garcia-Molina. PPay: micropayments for peer-to-peer systems. In *Proceedings of the 10th ACM conference on Computer and communications security (CCS '03)*, October 2003.
- [112] yufu office. What is local trading system yufu? Available electronically at <http://www.coara.or.jp/~yufukiri/letsyufu/> (*in Japanese*).