

VU Research Portal

Software Architecture Strategies for Cyber-Foraging Systems

Lewis, G.A.

2016

document version

Publisher's PDF, also known as Version of record

[Link to publication in VU Research Portal](#)

citation for published version (APA)

Lewis, G. A. (2016). *Software Architecture Strategies for Cyber-Foraging Systems*. [PhD-Thesis – Research external, graduation internal, Vrije Universiteit Amsterdam].

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:

vuresearchportal.ub@vu.nl

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Mobile Cloud Computing and Cyber-Foraging	2
1.3	Software Architecture and Cyber-Foraging	3
1.4	Research Questions	4
1.5	Thesis at a Glance	5
1.6	Research Methods	5
1.7	Outline of Thesis and Publications	7
2	A Systematic Literature Review of Architectural Design Decisions for Cyber-Foraging Systems	11
2.1	Research Protocol	11
2.1.1	Research Question	11
2.1.2	Search Strategy	12
2.1.3	Inclusion and Exclusion Criteria	12
2.1.4	Validation	12
2.2	Identification of Primary Studies	14
2.2.1	Round 1	14
2.2.2	Round 2	15
2.2.3	Final Round	15
2.3	Categorization of Primary Studies	23
2.3.1	Studies Per Type	23
2.3.2	Studies Per Year	23
2.4	Threats to Validity	25
2.5	Analysis of Primary Studies	25
2.5.1	Categorization of Architecture Decisions	25
2.5.1.1	Where to Offload	26
2.5.1.2	When to Offload	28
2.5.1.3	What to Offload	29
2.5.2	Analysis Results	32
2.5.2.1	Where to Offload	34
2.5.2.2	When to Offload	38
2.5.2.3	What to Offload	40
2.6	Main Observations and Findings from Primary Studies	51
2.7	Related Work	53
2.8	Summary and Conclusions	54

3	Architectural Tactics for Cyber-Foraging	57
3.1	Introduction	57
3.2	Functional Architectural Tactics for Cyber-Foraging	58
3.2.1	Computation Offload	60
3.2.2	Data Staging	63
3.2.2.1	Pre-Fetching	64
3.2.2.2	In-Bound Pre-Processing	67
3.2.2.3	Out-Bound Pre-Processing	69
3.2.3	Surrogate Provisioning	72
3.2.3.1	Pre-Provisioned Surrogate	72
3.2.3.2	Surrogate Provisioning from the Mobile Device	75
3.2.3.3	Surrogate Provisioning from the Cloud	77
3.2.4	Surrogate Discovery	80
3.2.4.1	Local Surrogate Directory	81
3.2.4.2	Cloud Surrogate Directory	83
3.2.4.3	Surrogate Broadcast	88
3.3	Non-Functional Architectural Tactics for Cyber-Foraging	92
3.3.1	Resource Optimization	92
3.3.1.1	Runtime Partitioning	92
3.3.1.2	Runtime Profiling	94
3.3.1.3	Resource-Adapted Computation	98
3.3.2	Fault Tolerance	101
3.3.2.1	Local Fallback	102
3.3.2.2	Opportunistic Mobile-Surrogate Data Synchronization	104
3.3.2.3	Cached Results	107
3.3.2.4	Alternate Communications	110
3.3.2.5	Eager Migration	114
3.3.3	Scalability/Elasticity	118
3.3.3.1	Just-in-Time Containers	118
3.3.3.2	Right-Sized Containers	120
3.3.3.3	Surrogate Load Balancing	122
3.3.4	Security	125
3.3.4.1	Trusted Surrogates	127
3.4	Summary and Conclusions	129

4	Case Study 1: Tactical Cloudlets — Cyber-Foraging for Computation Offload	131
4.1	Introduction	131
4.2	Case Study Design	132
4.2.1	Research Questions	132
4.2.2	Data Collection Procedure	132
4.2.3	Analysis Procedure	133
4.3	Results	133
4.3.1	System Context	133
4.3.2	System Requirements	134
4.3.2.1	Functional Requirements	134
4.3.2.2	Non-Functional Requirements	135
4.3.3	System Architecture and Design	135
4.3.4	Mapping of Architectural Design Decisions to Architectural Tactics	137
4.3.4.1	Computation Offload	138
4.3.4.2	Pre-Provisioned Surrogate	140
4.3.4.3	Surrogate Broadcast	142
4.3.4.4	Just-in-Time Containers	143
4.3.5	Analysis	145
4.3.5.1	Mapping between Tactics and Requirements	145
4.3.5.2	Discussion of Tactics for System Enhancements	149
4.3.5.3	Findings	151
4.3.6	Threats to Validity	153
4.4	Conclusions	153
4.5	Acknowledgments	154
5	Case Study 2: GigaSight — Cyber-Foraging for Data Staging	155
5.1	Introduction	155
5.2	Case Study Design	156
5.2.1	Research Questions	156
5.2.2	Data Collection Procedure	156
5.2.3	Analysis Procedure	156
5.3	Results	156
5.3.1	System Context	156
5.3.2	System Requirements	157
5.3.2.1	Functional Requirements	157
5.3.2.2	Non-Functional Requirements	157
5.3.3	System Architecture and Design	158

5.3.4	Mapping of Architectural Design Decisions to Architectural Tactics	160
5.3.4.1	Out-Bound Pre-Processing	160
5.3.4.2	Pre-Provisioned Surrogate	163
5.3.4.3	Local Surrogate Directory	165
5.3.4.4	Client-Side Data Caching	167
5.3.5	Analysis	167
5.3.5.1	Mapping between Tactics and Requirements	167
5.3.5.2	Discussion of Tactics for System Enhancements	172
5.3.5.3	Findings	173
5.3.6	Threats to Validity	175
5.4	Conclusions	175
5.5	Acknowledgments	176

6	Case Study 3: AgroTempus — Using Architectural Tactics for Cyber-Foraging Systems Development	177
6.1	Introduction	177
6.2	Case Study Design	178
6.2.1	Research Questions	178
6.2.2	Data Collection Procedure	178
6.2.3	Analysis Procedure	179
6.3	Results	180
6.3.1	System Context	180
6.3.2	System Requirements	181
6.3.2.1	Functional Requirements	181
6.3.2.2	Non-Functional Requirements	182
6.3.2.3	Constraints and Assumptions	185
6.3.3	Mapping of System Requirements to Architectural Tactics	185
6.3.3.1	Computation Offload	186
6.3.3.2	Out-Bound Pre-Processing	187
6.3.3.3	Pre-Fetching	187
6.3.3.4	Pre-Provisioned Surrogate	187
6.3.3.5	Surrogate Broadcast	188
6.3.3.6	Cached Results	188
6.3.3.7	Client-Side Data Caching	188
6.3.3.8	Just-in-Time Containers	189
6.3.4	System Architecture and Design	189
6.3.5	Mapping of Architectural Components to System Requirements	191

6.3.6	Mapping of Architectural Components to Identified Architectural Tactics	194
6.3.6.1	Computation Offload	194
6.3.6.2	Out-Bound Pre-Processing	194
6.3.6.3	Pre-Fetching	196
6.3.6.4	Pre-Provisioned Surrogate	199
6.3.6.5	Surrogate Broadcast	199
6.3.6.6	Cached Results	201
6.3.6.7	Client-Side Data Caching	203
6.3.6.8	Just-in-Time Containers	205
6.3.7	System Implementation	207
6.3.8	Analysis	209
6.3.8.1	System Evaluation	209
6.3.8.2	Developer Observation and Feedback	213
6.3.8.3	Findings	214
6.3.9	Threats to Validity	216
6.4	Conclusions	216
6.5	Acknowledgments	217
7	Characterization of Cyber-Foraging Usage Contexts	219
7.1	Introduction	219
7.2	Analysis	220
7.3	Cyber-Foraging Usage Contexts	223
7.4	Computation Offload Usage Contexts	225
7.4.1	Usage Context 1: Computation-Intensive Mobile Applications (Short Operations)	225
7.4.2	Dynamic Environments	227
7.4.2.1	Usage Context 2: Mobile Applications in Low Coverage Environments.	228
7.4.2.2	Usage Context 3: Computation-Intensive Mobile applications (Long Operations).	229
7.4.2.3	Usage Context 4: Computation-Intensive Mobile Applications in Hostile Environments.	231
7.4.2.4	Usage Context 5: Public Surrogates.	233
7.5	Data Staging Usage Contexts	235
7.5.1	Usage Context 6: Sensing Applications	235
7.5.2	Usage Context 7: Data-Intensive Mobile Applications	237
7.6	Summary and Conclusions	239

8	Decision Model for Cyber-Foraging Systems	243
8.1	Introduction	243
8.2	Mapping the Problem Space to the Solution Space	244
8.3	How to Use the Decision Models	246
8.4	Decision Models for Cyber-Foraging Systems	248
8.4.1	Data Staging	248
8.4.2	Surrogate Provisioning	251
8.4.3	Surrogate Discovery	253
8.4.4	Resource Optimization	256
8.4.5	Fault Tolerance	258
8.4.6	Scalability and Elasticity	262
8.4.7	Security	264
	8.4.7.1 Credential Exchange	264
	8.4.7.2 Credential Validation	266
8.5	Validation	266
8.6	Related Work	268
8.7	Conclusions	269
9	Conclusions	271
9.1	Contributions	271
9.1.1	RQ1: What Software Architecture Design Decisions for Cyber-Foraging Systems can be Identified in the Literature?	272
9.1.2	RQ2: What Architectural Tactics can be Derived from the Identified Architectural Design Decisions?	273
9.1.3	RQ3: What are the Usage Domains and Contexts (Defined in Terms of Functional and Non-Functional Requirements) that Benefit from Cyber-Foraging?	274
9.1.4	RQ4: How to Support Architectural Design Decision Making in Cyber-Foraging Systems?	274
9.2	Future Research	275
9.2.1	Extension of the Tactics Catalog	275
9.2.2	Quantitative Analysis of the Impact of Tactics Selection	276
9.2.3	Tools for the Development and Analysis of Cyber-Foraging Systems	277
9.2.4	Architecture Patterns for Cyber-Foraging Systems	278