Analytics of tools and methodologies for blockchain-based software systems development

Master’s thesis presentation

Minh-Duc Ta

23.07.2020

Aalto University
School of Science

Outline

- Introduction
- Background knowledge
- Analyses
- Blockchain’s PrAgmatic Knowledge Explorer (BAKE)
- Evaluation
- Conclusions

Introduction

Introduction

- **Blockchain**
- **Blockchain-based project**
    - ”- is any project that consists of any activity directly relating to or indirectly relating to blockchain technology”
- **Motivations**
    - Increasing adoption but the development is still slow
    - Business demands

Research topics

- Analytics of tools and methodologies in BBP
- Knowledge collecting tool providing insights on BBP

Contributions

- Overview picture and analyses of the software development in BBP
- Blockchain’s PrAgmatic Knowl-edge Explorer (BAKE)

Background & related works

Blockchain overview

- **Blockchain overview**
- **Common challenges**

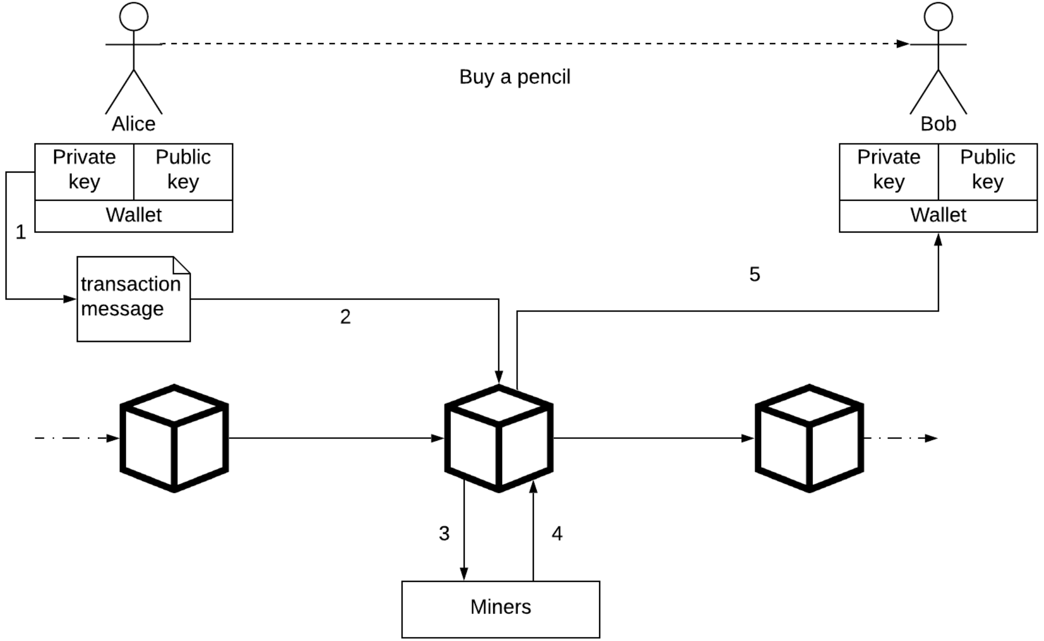


Figure 1: How blockchain works in general

Software development in BBP

- **Stakeholders in software development life cycle (SDLC)**
- **Current researches on software development in BPP**
    - Typical problems
    - Development practices
    - Development process improvements
- **Researching gaps**
    - Usage of tools
    - Limitation and practices using tools
    - Linking to development activities

Analyses

Objectives

- Comparison of development roles in BBP and normal projects.
- Comparison of the usage of programming languages, development tools and frameworks between BBP and normal projects.
- Usage of development tools in real-life BBP in SDLC

Research approaches (1)

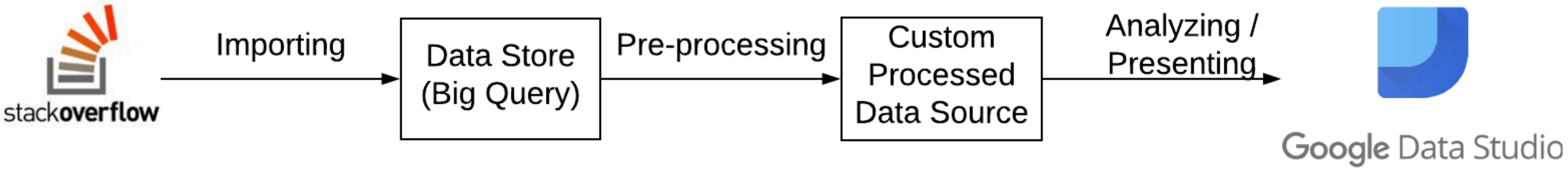


Figure 2: Analysing process for raw public data set

Research approaches (2)

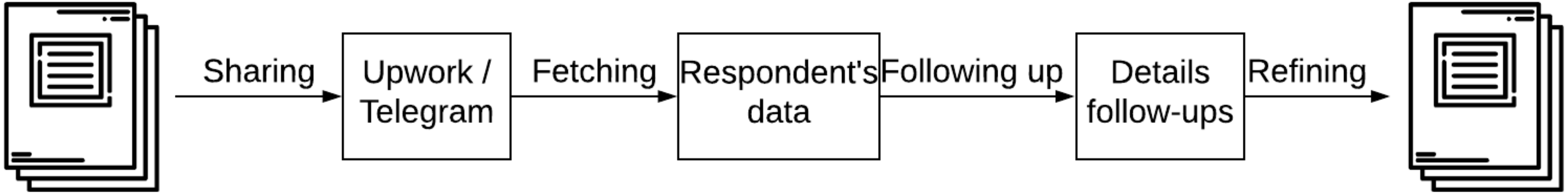
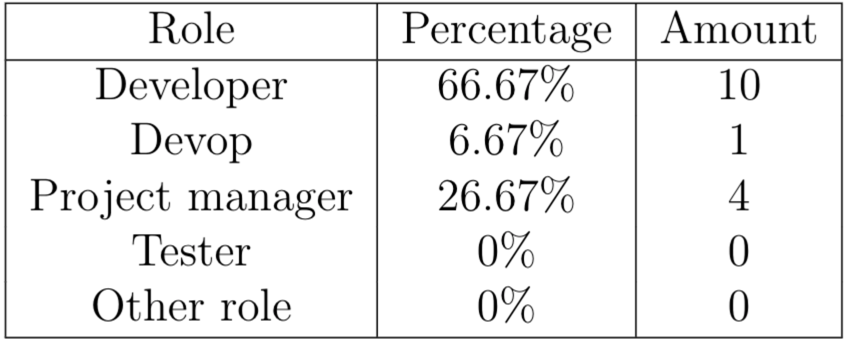


Figure 3: Analysing process for own conducting survey

Analytics

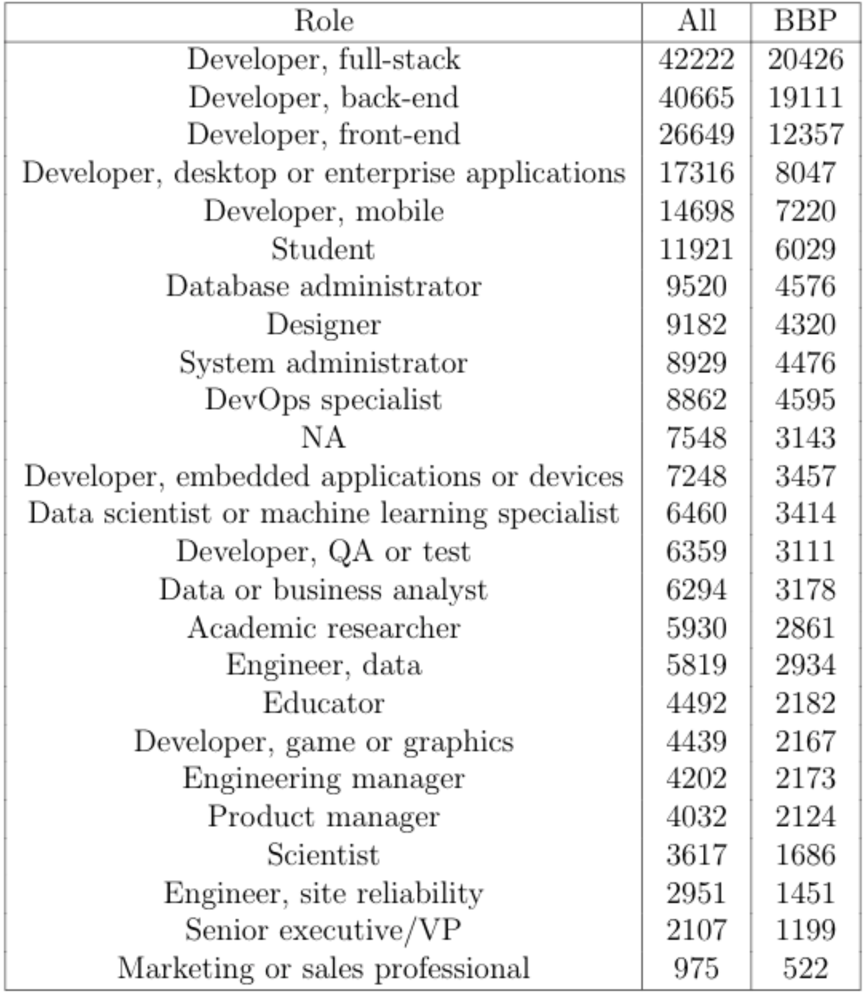
- Stakeholders in SDLC (1) (2)
- Programming languages (1) (2)
- Development platform (1)
- Development environment (1)
- Usage of frameworks and tools (1) (2)
- Limitations and challenges in BBP (2)

Analytics - stakeholders



Role	Percentage	Amount
Developer	66.67%	10
Devop	6.67%	1
Project manager	26.67%	4
Tester	0%	0
Other role	0%	0

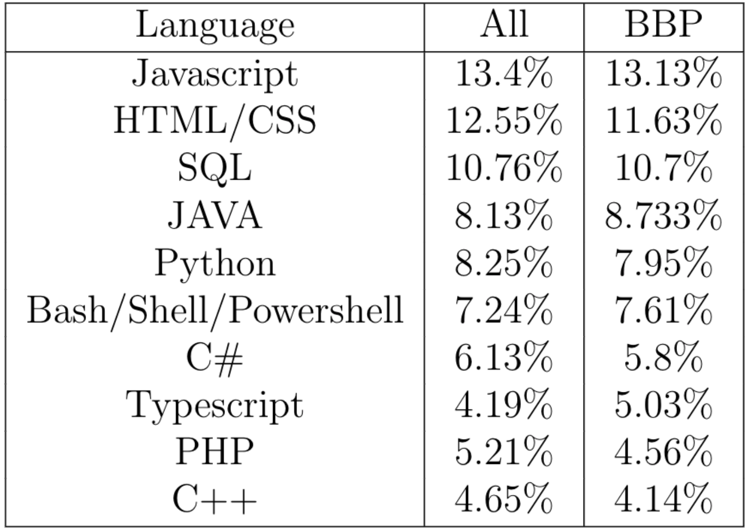
Table 1. Main roles in own conducting survey



Role	All	BBP
Developer, full-stack	42222	20426
Developer, back-end	40665	19111
Developer, front-end	26649	12357
Developer, desktop or enterprise applications	17316	8047
Developer, mobile	14698	7220
Student	11921	6029
Database administrator	9520	4576
Designer	9182	4320
System administrator	8929	4476
DevOps specialist	8862	4595
NA	7548	3143
Developer, embedded applications or devices	7248	3457
Data scientist or machine learning specialist	6460	3414
Developer, QA or test	6359	3111
Data or business analyst	6294	3178
Academic researcher	5930	2861
Engineer, data	5819	2934
Educator	4492	2182
Developer, game or graphics	4439	2167
Engineering manager	4202	2173
Product manager	4032	2124
Scientist	3617	1686
Engineer, site reliability	2951	1451
Senior executive/VP	2107	1199
Marketing or sales professional	975	522

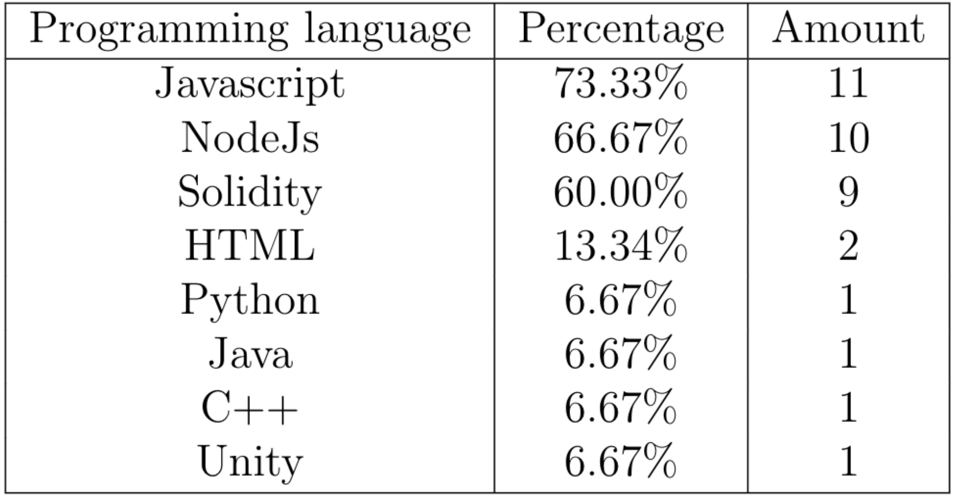
Table 2. Popular roles from aggregated public data set

Analytics - languages



Language	All	BBP
Javascript	13.4%	13.13%
HTML/CSS	12.55%	11.63%
SQL	10.76%	10.7%
JAVA	8.13%	8.733%
Python	8.25%	7.95%
Bash/Shell/Powershell	7.24%	7.61%
C#	6.13%	5.8%
Typescript	4.19%	5.03%
PHP	5.21%	4.56%
C++	4.65%	4.14%

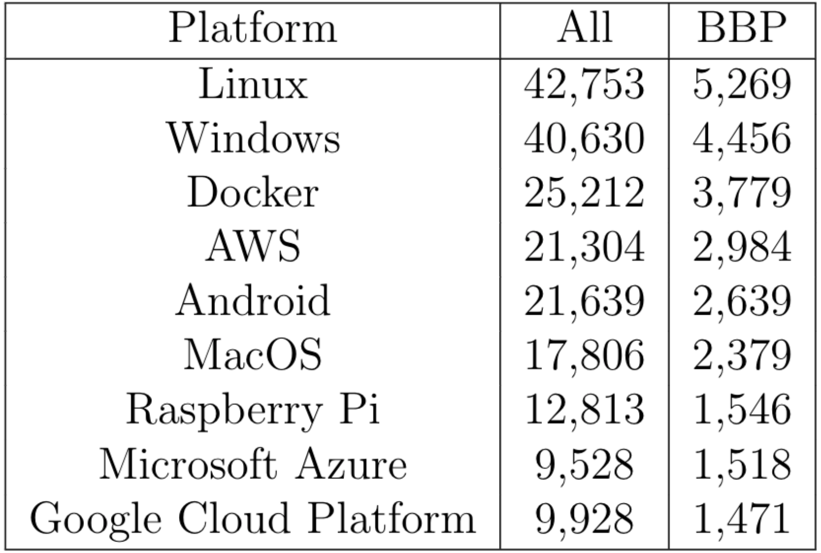
Table 3. Most used languages aggregated from public data set



Programming language	Percentage	Amount
Javascript	73.33%	11
NodeJs	66.67%	10
Solidity	60.00%	9
HTML	13.34%	2
Python	6.67%	1
Java	6.67%	1
C++	6.67%	1
Unity	6.67%	1

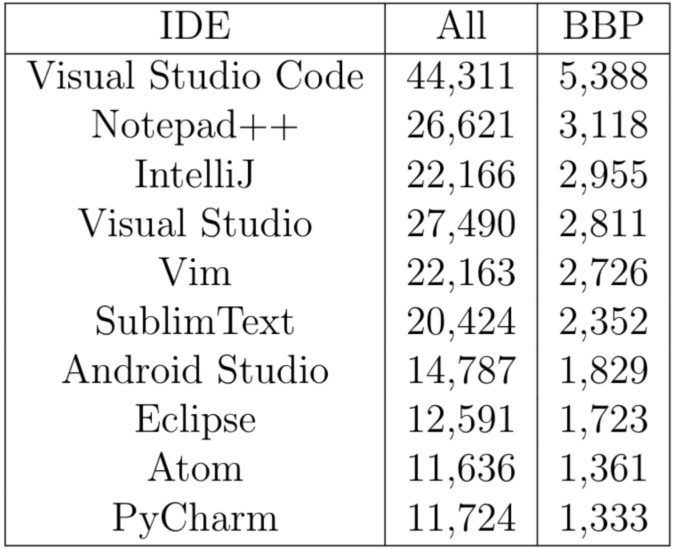
Table 4. Used languages in projects recorded in own conducting survey

Analytics - dev environment



Platform	All	BBP
Linux	42,753	5,269
Windows	40,630	4,456
Docker	25,212	3,779
AWS	21,304	2,984
Android	21,639	2,639
MacOS	17,806	2,379
Raspberry Pi	12,813	1,546
Microsoft Azure	9,528	1,518
Google Cloud Platform	9,928	1,471

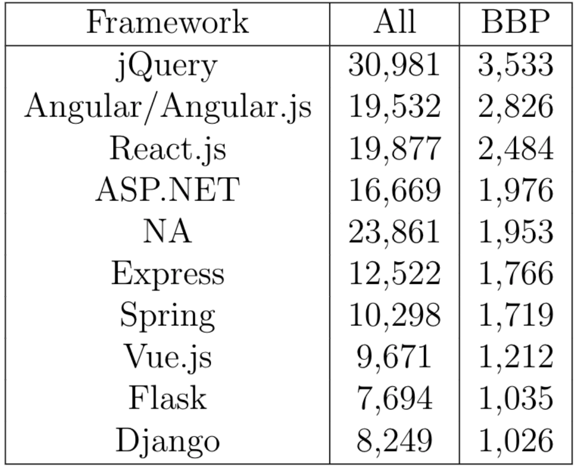
Table 5. Most used development platform aggregated from public data set



IDE	All	BBP
Visual Studio Code	44,311	5,388
Notepad++	26,621	3,118
IntelliJ	22,166	2,955
Visual Studio	27,490	2,811
Vim	22,163	2,726
SublimText	20,424	2,352
Android Studio	14,787	1,829
Eclipse	12,591	1,723
Atom	11,636	1,361
PyCharm	11,724	1,333

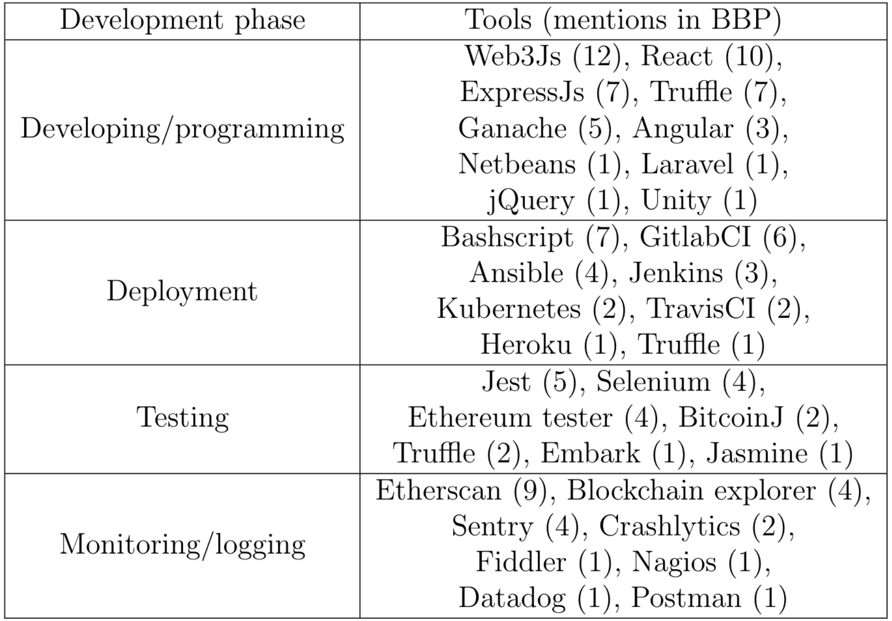
Table 6. Most used development environment aggregated from public data set

Analytics – tools & frameworks



Framework	All	BBP
jQuery	30,981	3,533
Angular/Angular.js	19,532	2,826
React.js	19,877	2,484
ASP.NET	16,669	1,976
NA	23,861	1,953
Express	12,522	1,766
Spring	10,298	1,719
Vue.js	9,671	1,212
Flask	7,694	1,035
Django	8,249	1,026

Table 7. Most used web-frameworks aggregated from public data set



Development phase	Tools (mentions in BBP)
Developing/programming	Web3Js (12), React (10), ExpressJs (7), Truffle (7), Ganache (5), Angular (3), Netbeans (1), Laravel (1), jQuery (1), Unity (1)
Deployment	Bashscript (7), GitlabCI (6), Ansible (4), Jenkins (3), Kubernetes (2), TravisCI (2), Heroku (1), Truffle (1)
Testing	Jest (5), Selenium (4), Ethereum tester (4), BitcoinJ (2), Truffle (2), Embark (1), Jasmine (1)
Monitoring/logging	Etherscan (9), Blockchain explorer (4), Sentry (4), Crashlytics (2), Fiddler (1), Nagios (1), Datadog (1), Postman (1)

Table 8. Tools and frameworks used in projects in own conducting survey

Analytics - challenges

- **Technical challenges**
  - production-staging differences
  - Non-supported functionality
- **Non-technical challenges**
  - Complexity and conflict of tool configurations
  - Tool documentation

Analytics - conclusion

- **Overview of usage of tools**
- **Limitations**
    - Relationship between tools
    - Relationship between tools, blockchain and other components
    - The analytic data can easily be outdated

Blockchain’s PrAgmatic Knowledge Explorer (BAKE)

BAKE

To improve the analytics to generate a better understanding of thesoftware development in BBP as well as automate the knowledge collectingand generating processes.

- ❑ **Centralized storage of knowledge**
- ❑ **Data collection and data fetching**
- ❑ **Simple bootstrapping pipelines of knowledge**
- ❑ **Interactive knowledge exploration**

System architecture

Main developments:

- Open Rest API
- serialization in ORM layer
- Command routing in KRCR layer for BAKE APIs
- BAKE CLI

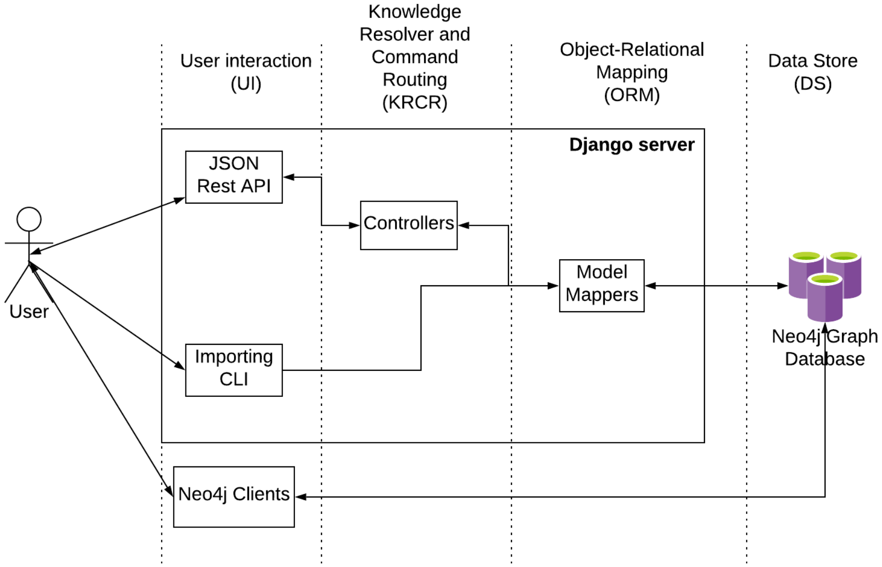


Figure 4: BAKE system architecture

Aalto University
School of Science

Knowledge design

Groups of knowledge

- Application and project information
- Activity information
- Tools and their relationships with others
- Blockchain and its connection with different categories

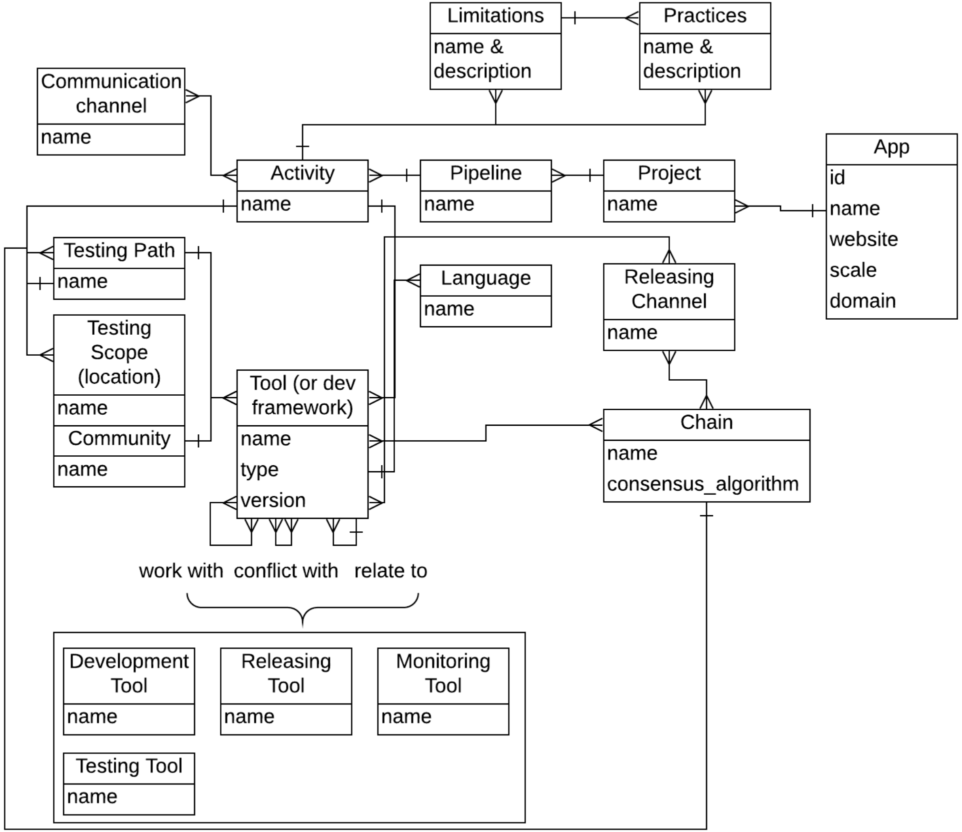


Figure 5: Knowledge schema

Aalto University
School of Science

Data boostrapping

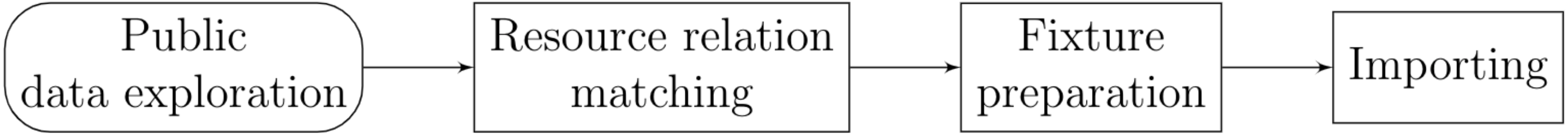

Public data exploration → Resource relation matching → Fixture preparation → Importing

Figure 6: Generic knowledge resource boostrapping

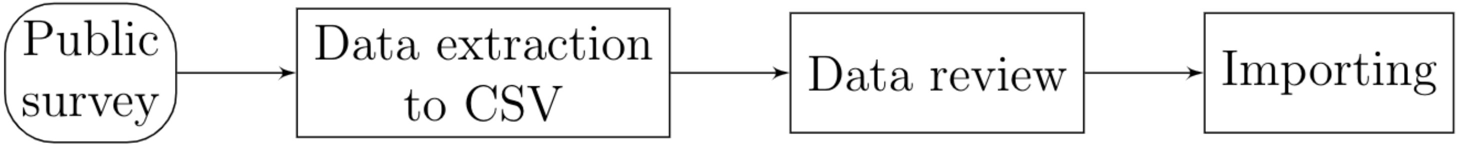

Public survey → Data extraction to CSV → Data review → Importing

Figure 7: Overall resource boostrapping

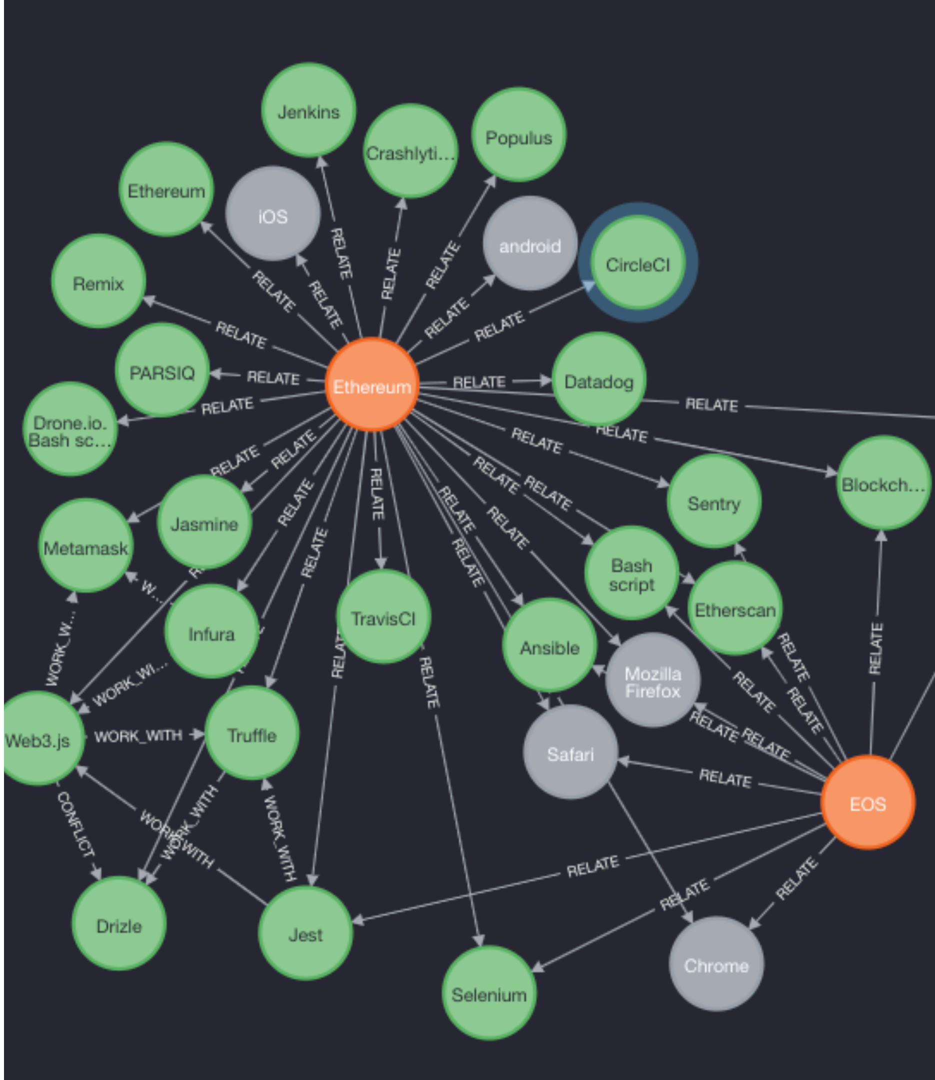
Evaluation

Setup

Deployment to dedicated virtual machine:

- 1 GB Memory,25 GB Disk, Ubuntu 18.04.3 (LTS) x64.
- Nginx 1.14.0 as web server
- Neo4J 3.5.14 as graph database entity
- Python 3.8.1 as executing environment of the application
- Gunicorn 20.0.4 as Python WSGI HTTP Server
- 114 node records and 191 relationship records boostrapped

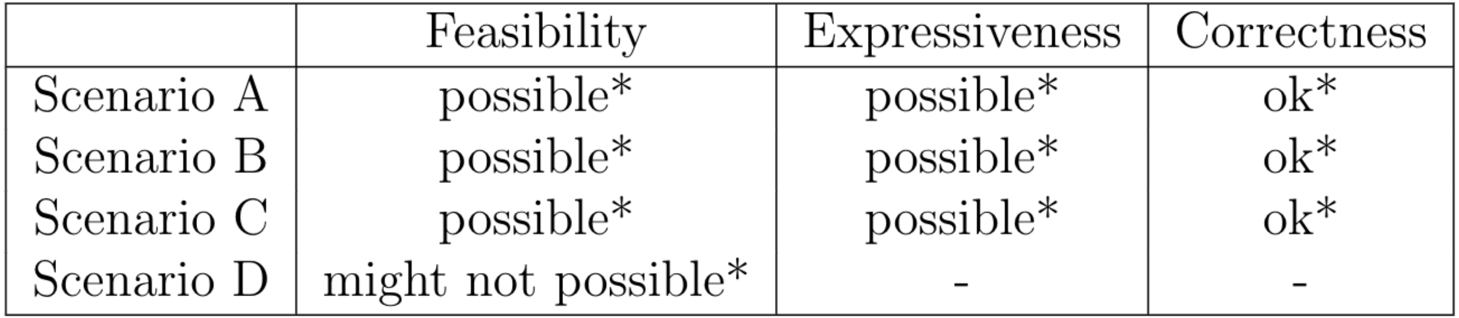
DEMO


Jenkins
Crashlyti…
Populus
Ethereum
iOS
android
CircleCI
Remix
PARSIQ
Drone.io. Bash sc…
Ethereum
Datadog
Blockch…
Sentry
Metamask
Jasmine
Bash script
Etherscan
Infura
TravisCI
Ansible
Mozilla Firefox
Web3.js
Truffle
Safari
EOS
Drizle
Jest
Selenium
Chrome
RELATE
WORK_WITH
CONFLICT

Scenario samples

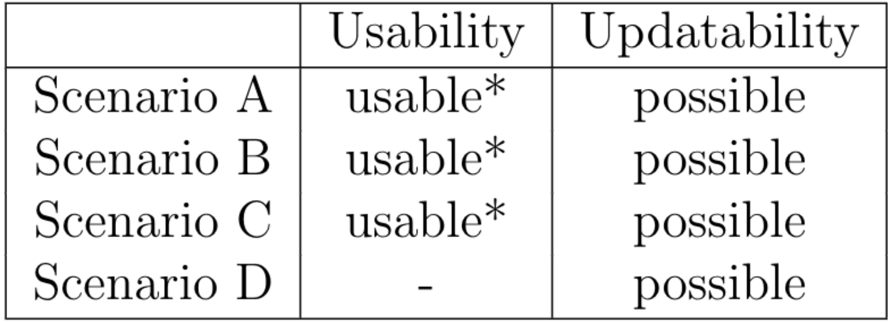
- Scenario A: a developer
- Scenario B: a senior devop
- Scenario C: a senior tester
- Scenario D: a designer

Criteria



	Feasibility	Expressiveness	Correctness
Scenario A	possible*	possible*	ok*
Scenario B	possible*	possible*	ok*
Scenario C	possible*	possible*	ok*
Scenario D	might not possible*	-	-

Table 9: Evaluation of scenarios based on functional parameters



	Usability	Updatability
Scenario A	usable*	possible
Scenario B	usable*	possible
Scenario C	usable*	possible
Scenario D	-	possible

Table 10: Evaluation of scenarios based on non-functional parameters

Performance measurement

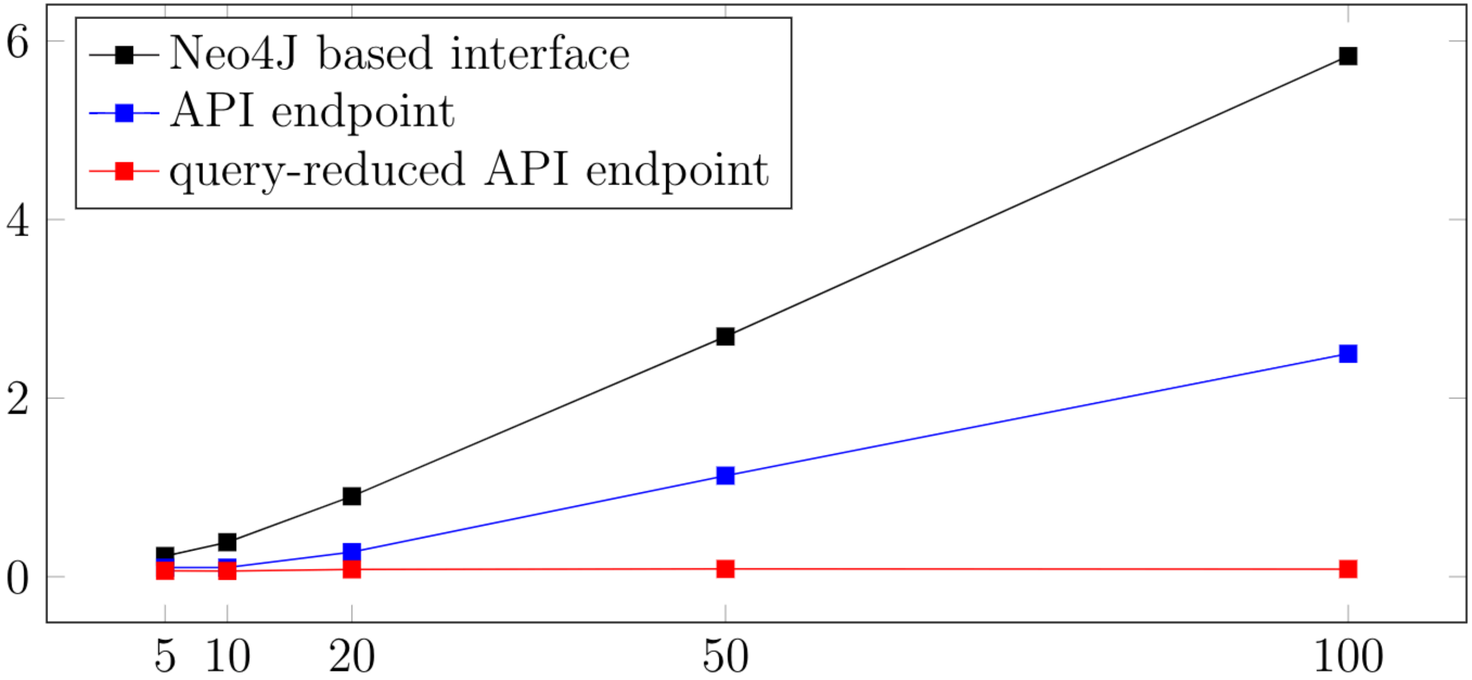


Figure 8: Average response time (in seconds) of requests to Neo4J-basedand API interface

Aalto University
School of Science

Conclusions

Limitations and future work

- **Limitations**
    - limited types of stakeholders
    - limited generated knowledge
- **Further works**
    - automation and collection from different sources
    - Identify the demands from views of other stakehoders
    - deeper knowledge collection