Automated documentation of Business Domain assignments and cloud application information from an application development pipeline

Master Thesis: Kickoff

Nicolas Corpancho Villasana 10.09.2018

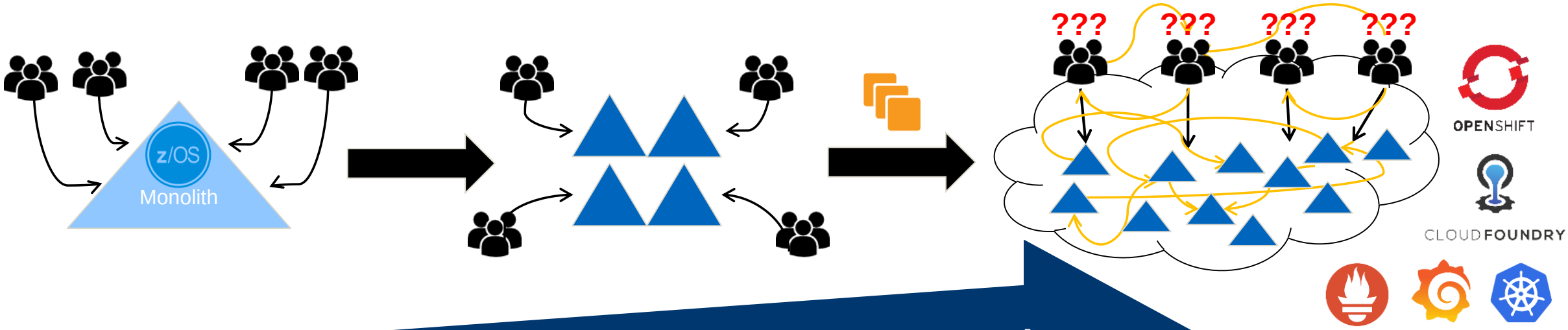
Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Agenda

1. Motivation
2. Literature review
3. Derivation for EAM
4. Research questions
5. Evaluation environment
6. Solution
7. Timeline
8. Next steps

1. Motivation: Major technology trends that have an impact on EAM

1. **Agile development** and **continuous delivery (CD)** approaches are becoming more important in todays enterprises. CD leads to **short lifecycles** which implies a **quicker update of the EA Tool information**
2. **Cloud migration** reduces infrastructure and maintenance costs. **Cloud reduces transparency** and increases **complexity of EA documentation**
3. **Modularization** of **legacy systems** into **application components and microservices** causes a **higher number of applications**



2. Literature review

Year	Author	Title	Outlook
2010	Matthias Farwick, Berthold Agreiter, Ruth Breu, Matthias Häring, Karsten Voges, Inge Hanschke	Towards Living Landscape Models: Automated Integration of Infrastructure Cloud in Enterprise Architecture Management	Cloud Integration (PaaS and SaaS)
2012	Buschle, M., Ekstedt, M., Grunow, S., Matheus Hauder, Florian Matthes, Sascha Roth	Automated Enterprise Architecture Documentation using an Enterprise Service Bus	Modeling EA transformations
2013	Matthias Farwick, Ruth Breu, Matheus Hauder, Sascha Roth, Florian Matthes	Enterprise Architecture Documentation: Empirical Analysis of Information Sources for Automation	Automation opportunities
2013	Sascha Roth, Matheus Hauder, Felix Michel, Dominik Münch, Florian Matthes	Facilitating Conflict Resolution of Models for Automated Enterprise Architecture Documentation	Combination of approach with other automated EA documentation approaches
2013	Sascha Roth, Matheus Hauder, Matthias Farwick, Ruth Breu, and Florian Matthes	Enterprise Architecture Documentation: Current Practices and Future Directions	Automation mechanisms to improve EA documentation
2018	J org Landthaler, Ömer Uludag, Gloria Bondel, Ahmed Elnaggar, Saasha Nair, and Florian Matthes	A Machine Learning Based Approach to Application Landscape Documentation	Research in automated EAD

Research in automated EA documentation integrating todays Cloud (PaaS and SaaS) is needed

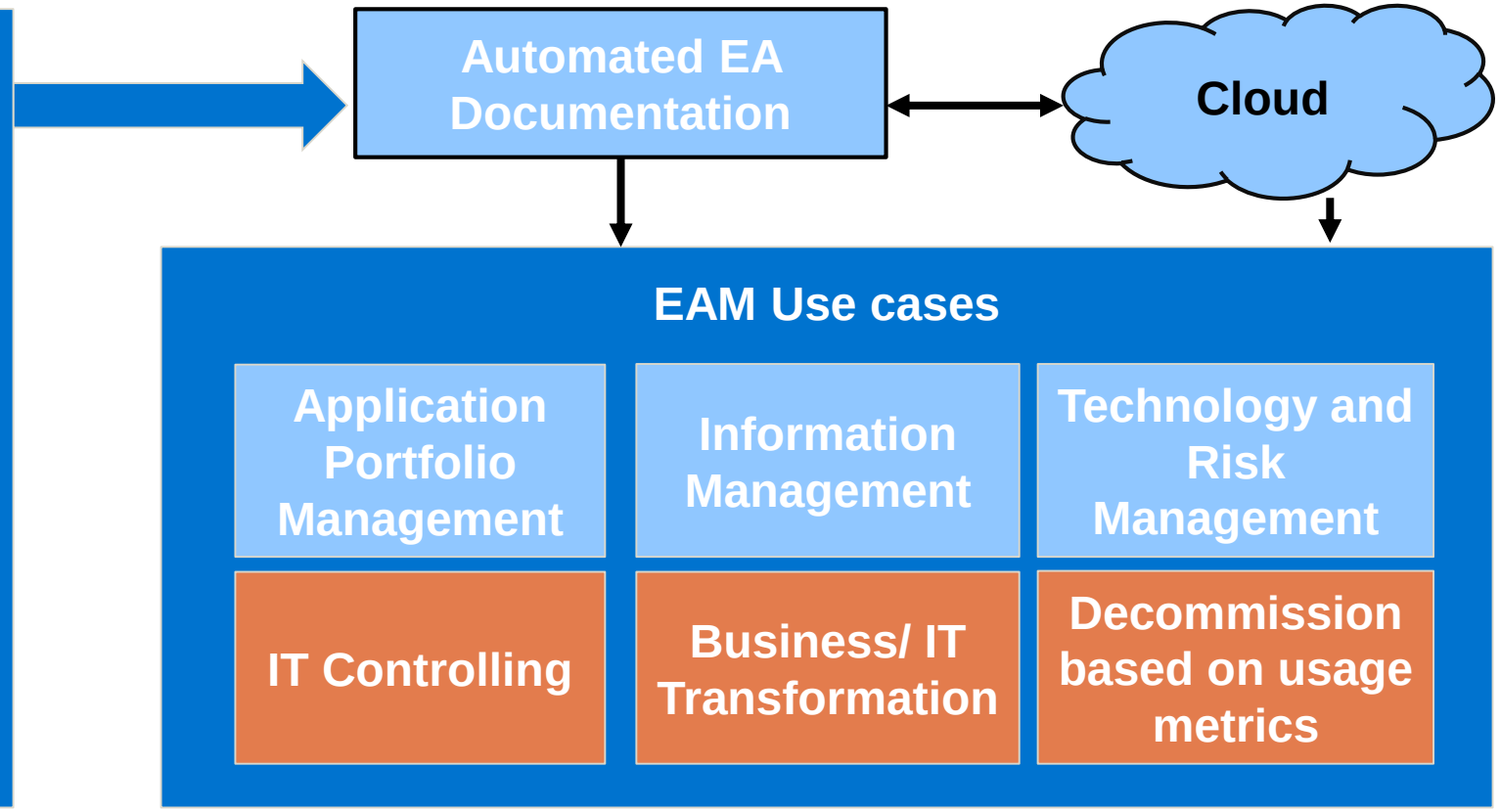
Derivation for EAM

Challenges of current EA documentation

- No single automated EA documentation standard
- Many documenting approaches exist, mostly manual data collection
- Helps to manage information complexity due to higher number of applications in different environments
- Increases support fast changing environments (multi environments) and technologies

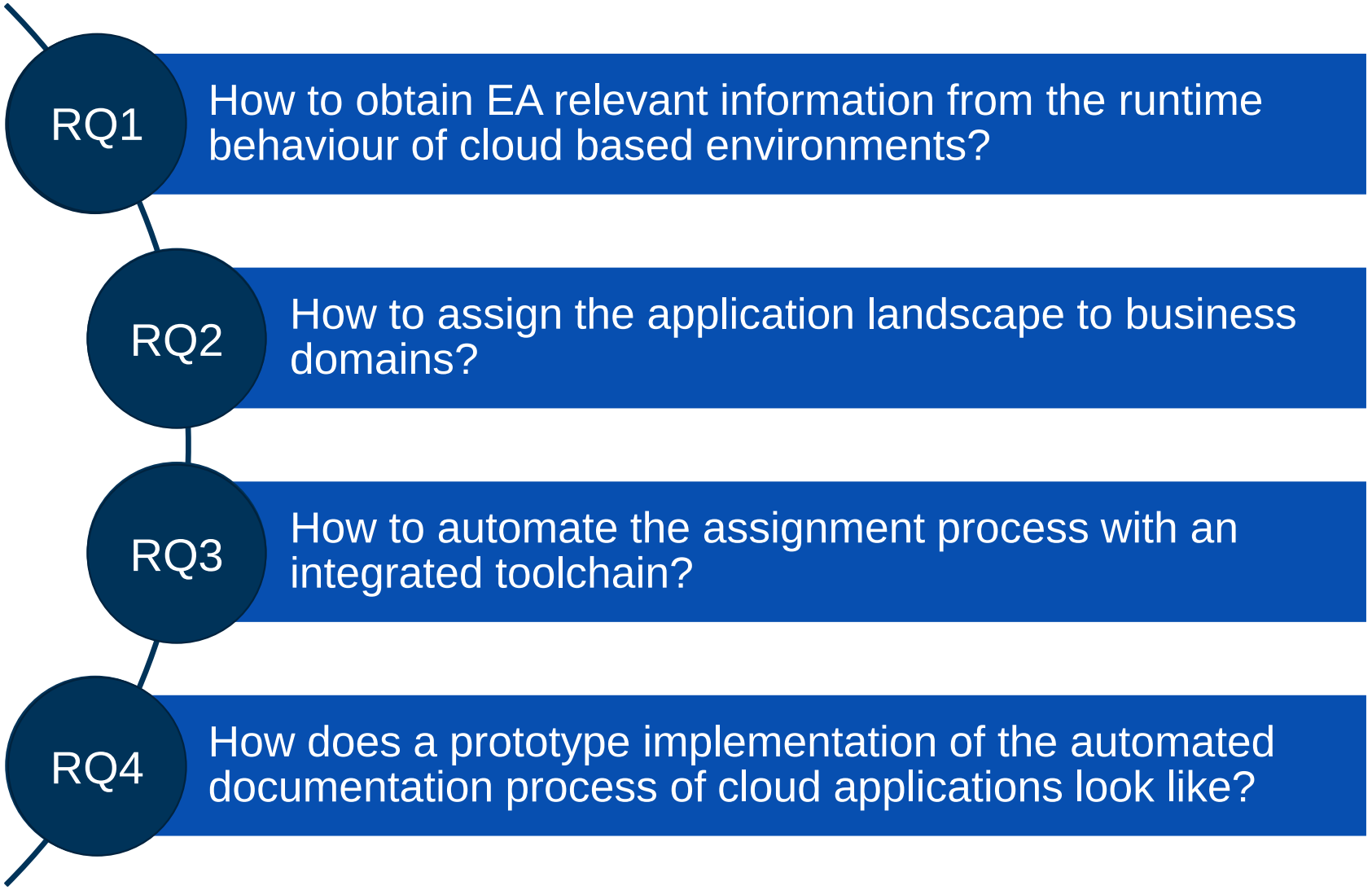
Consequences

- High error rate documenting and collecting EA information
- Time consuming collection of information
- Expensive tasks maintaining and gathering information of the EA

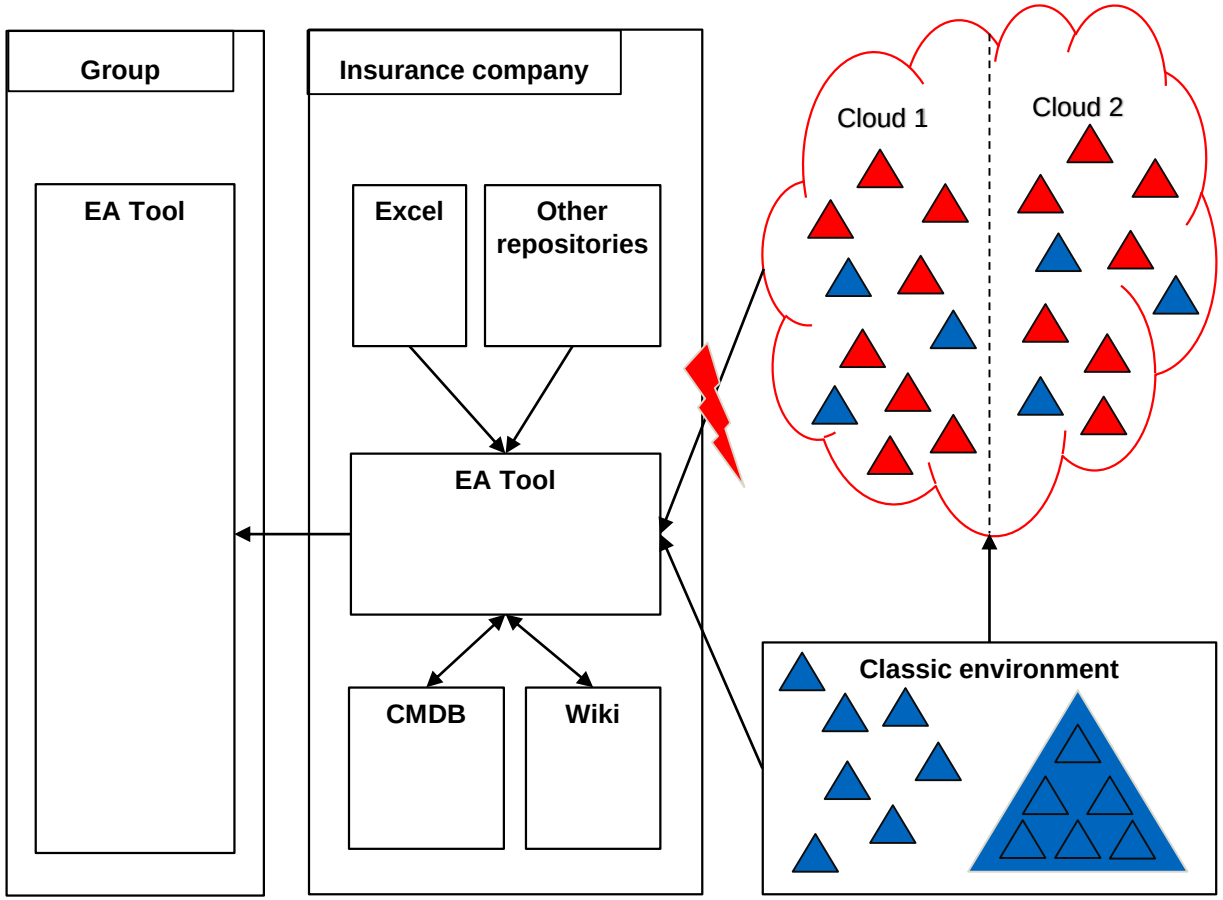


Improved EAM use cases can be derived from an automated EA documentation process

4. Research Questions



5. Evaluation environment

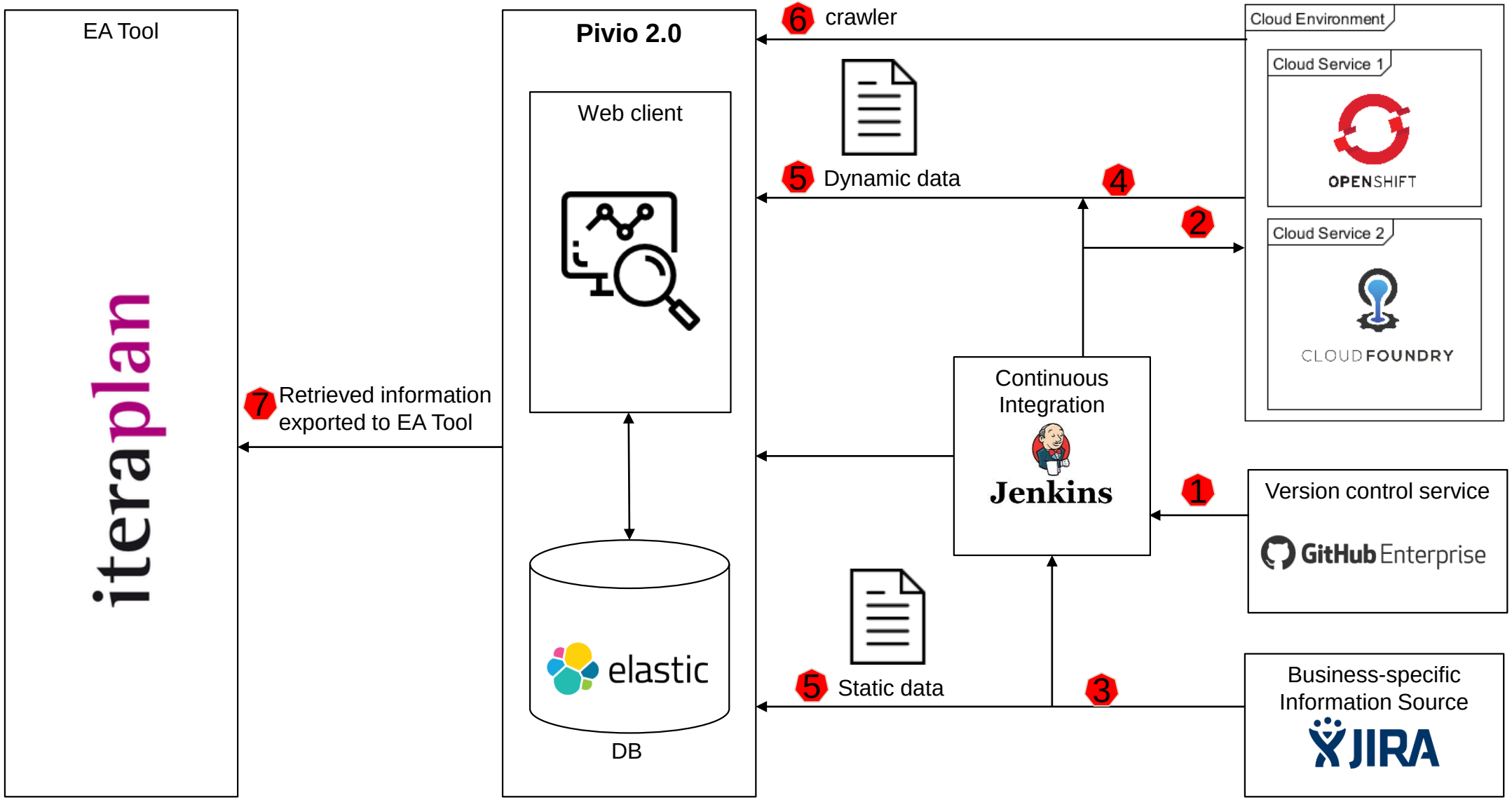


- Manual collection of EA information with Excel
- Integration of other repositories into EA Tool
- Import of Excel into EA Tool
- Federated approach relating EA Tool to other Tools such as a CMDB and a Wiki
- Integration of application inventory to Group EA Tool
- No cloud application documentation
- Modularization of legacy systems into application components and microservices
- Migration of application components and microservices to cloud environments

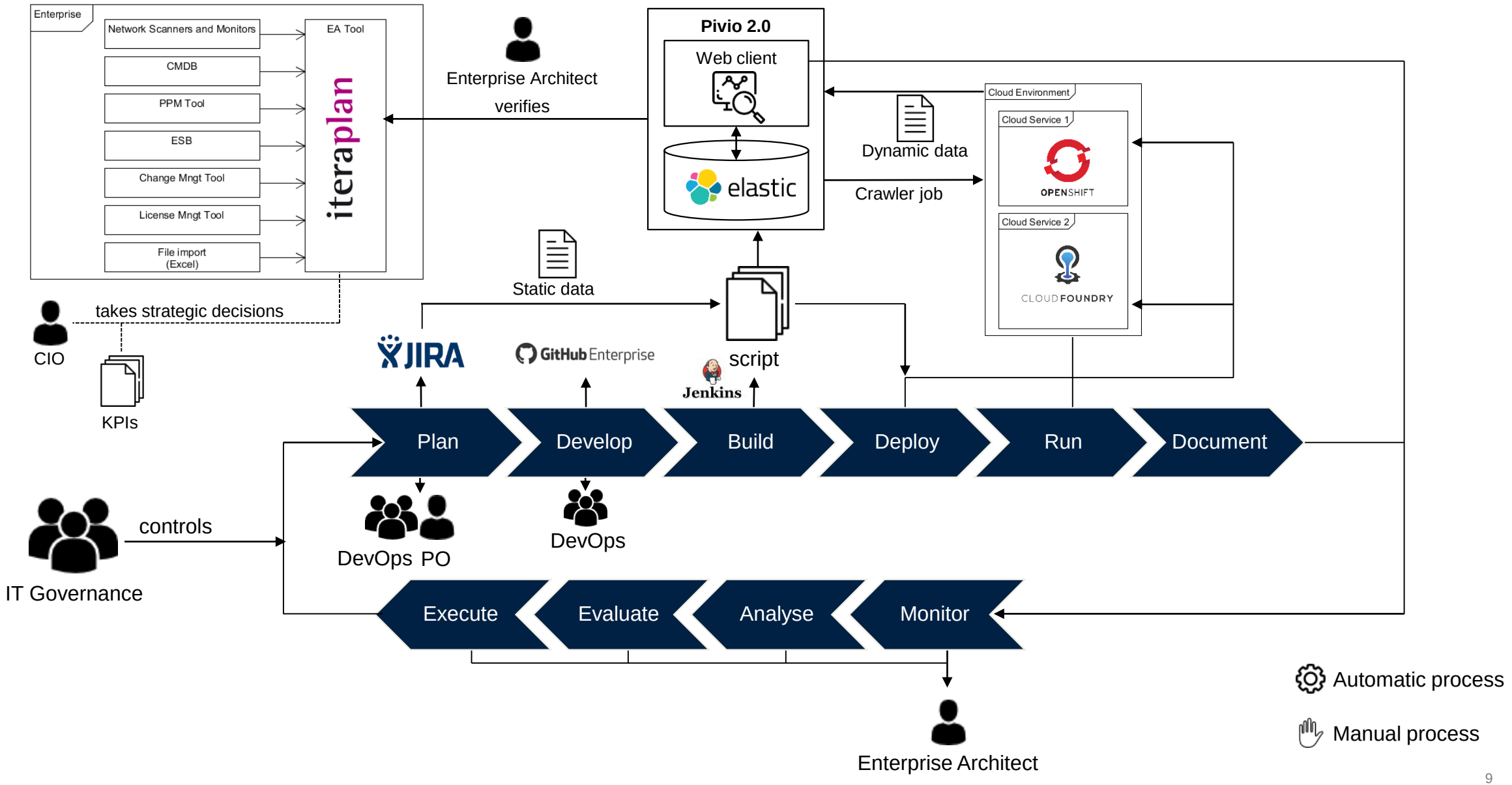
Non documented applications

Documented applications

6. Solution: High level architecture



6. Solution: Information collection process

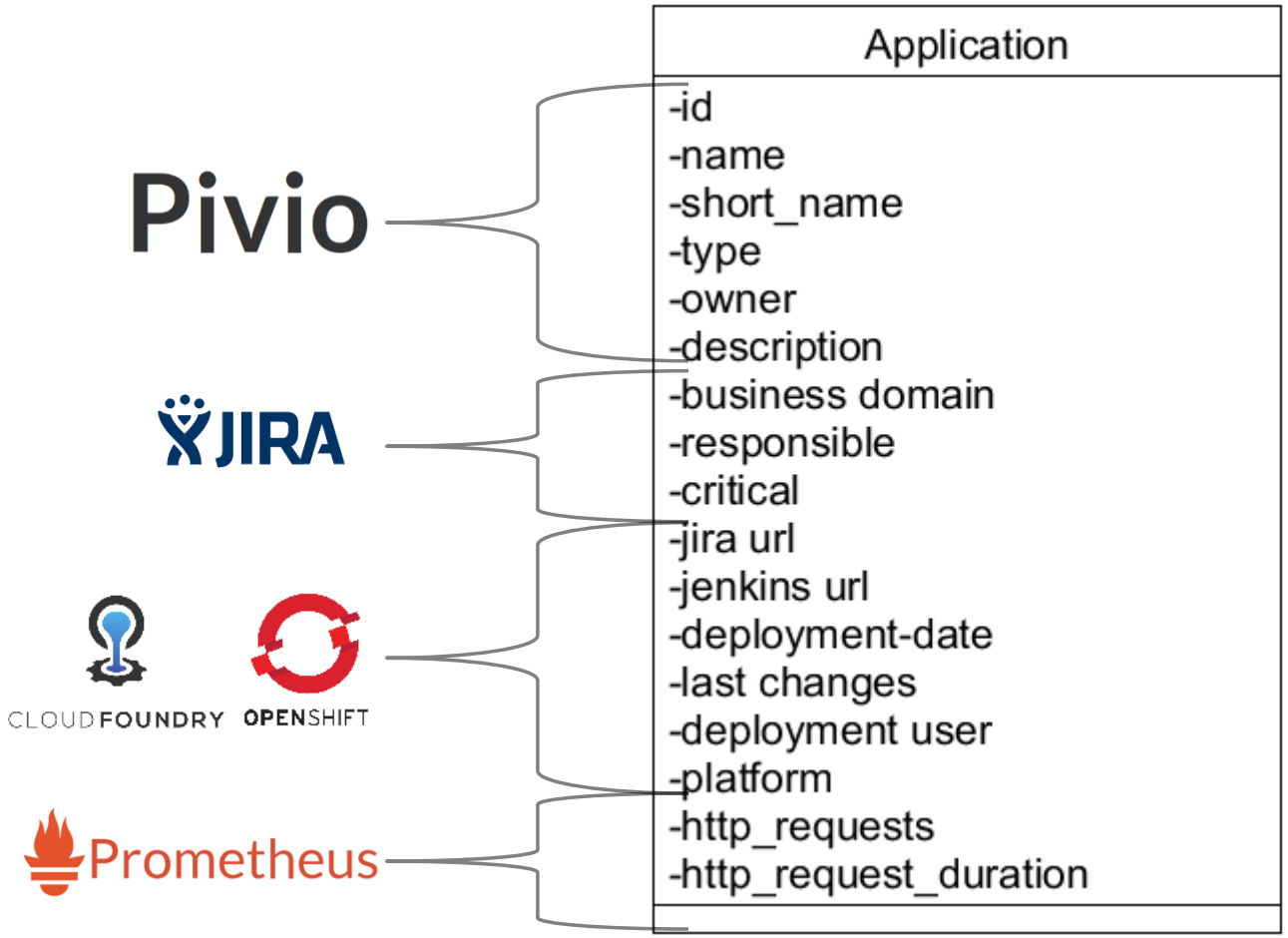


6. Solution: Data model

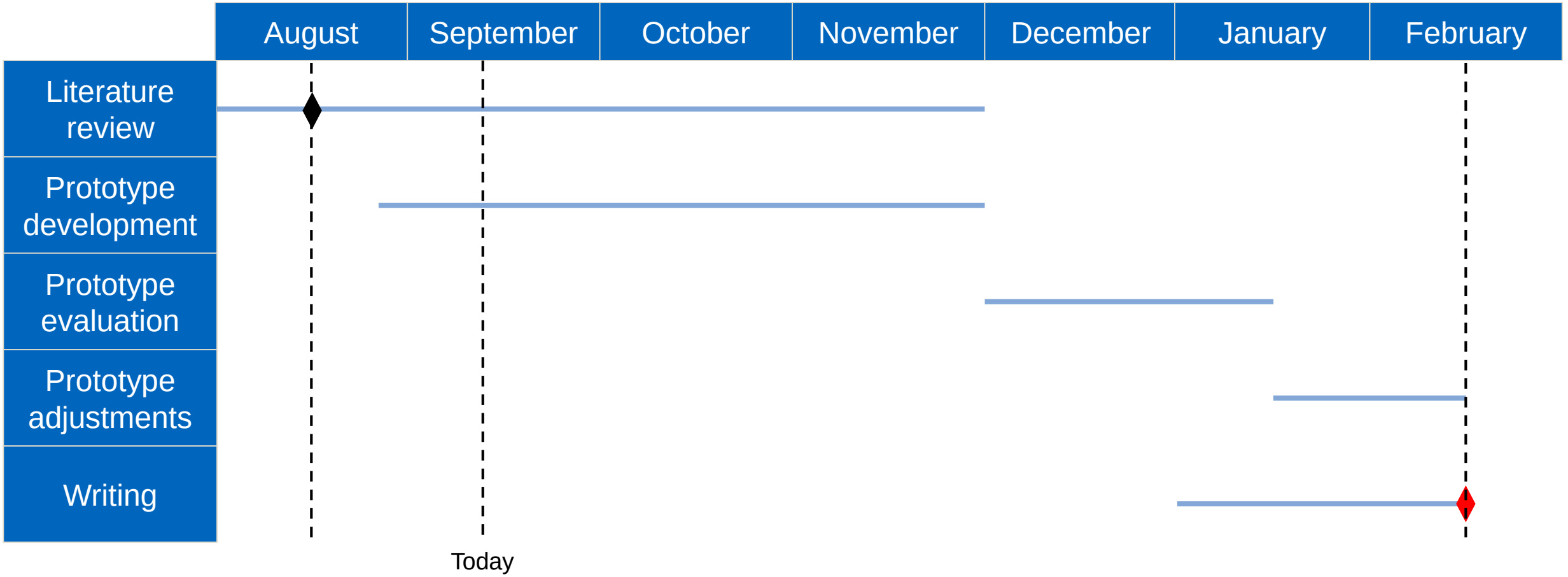
- Attributes structured in 3 parts:
  1. Required fields in Pivio
  2. Business specific information
  3. Technical specifications
  4. Application performance metrics

1. Required fields from data model of Pivio
2. Business specific information is gathered from JIRA
3. Technical information is retrieved from the cloud environments
4. Application performance metrics collected from monitoring systems such as Prometheus

➢ Attributes extend existing "*Anwendung-Service (IS)*"*-object* in EA Tool (iteraplan)

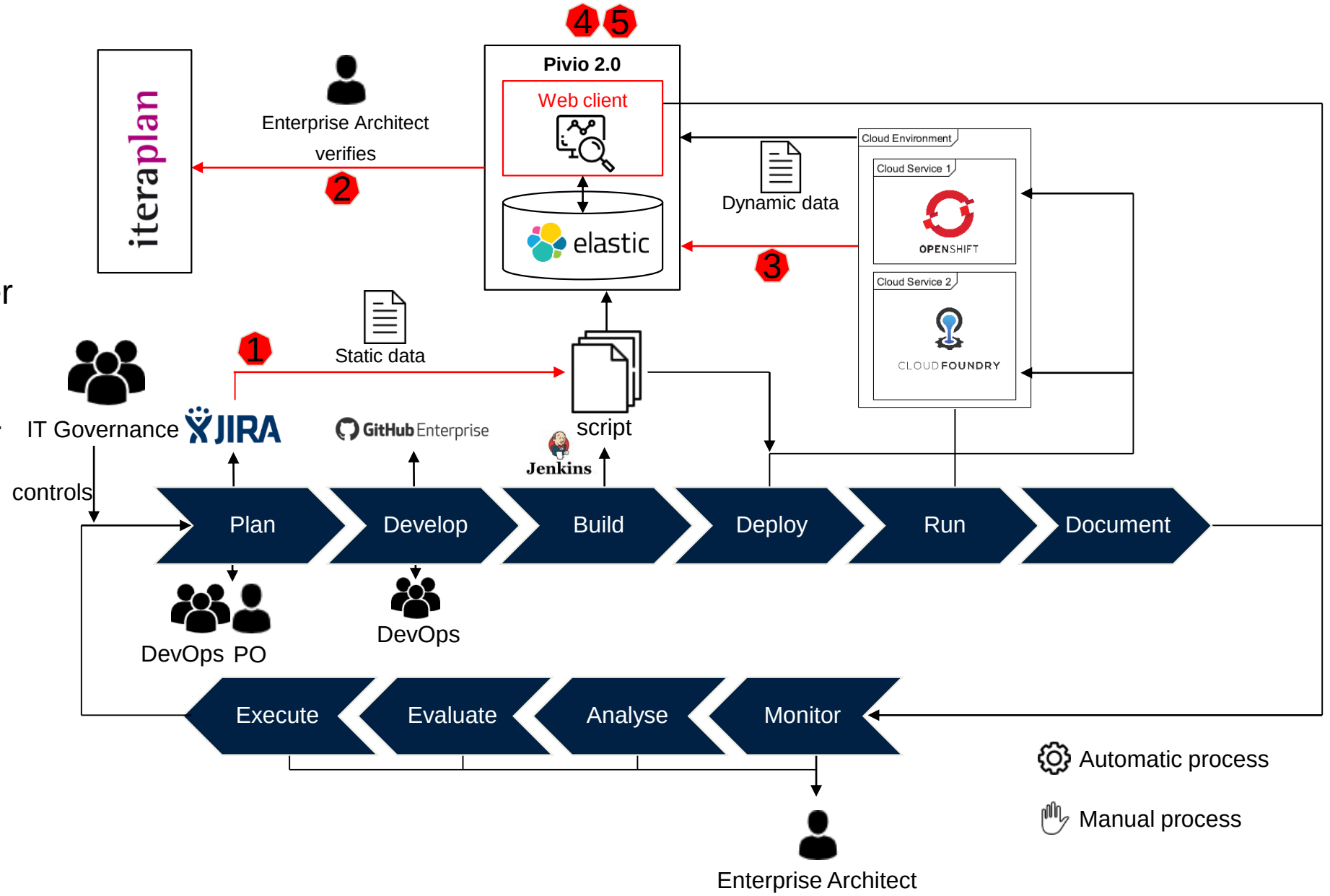


7. Timeline

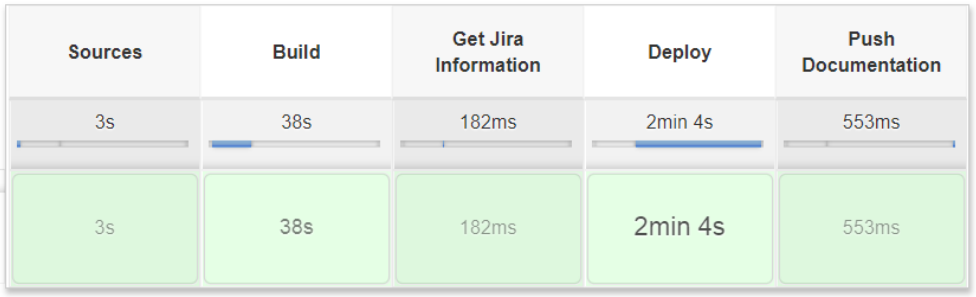

August
September
October
November
December
January
February
Literature review
Prototype development
Prototype evaluation
Prototype adjustments
Writing
Today
Registered Date: 15.08.2018
Submission Date: 15.02.2019

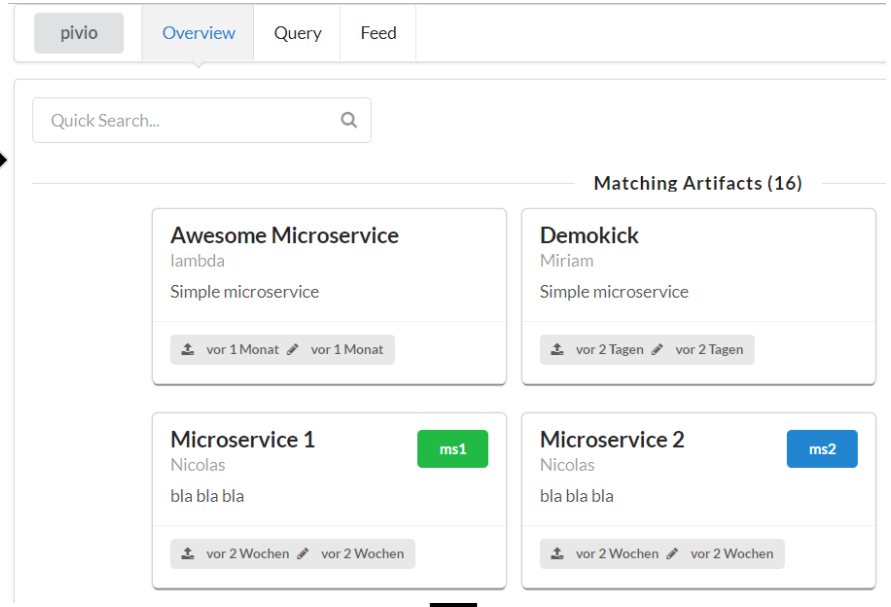
8. Next Steps

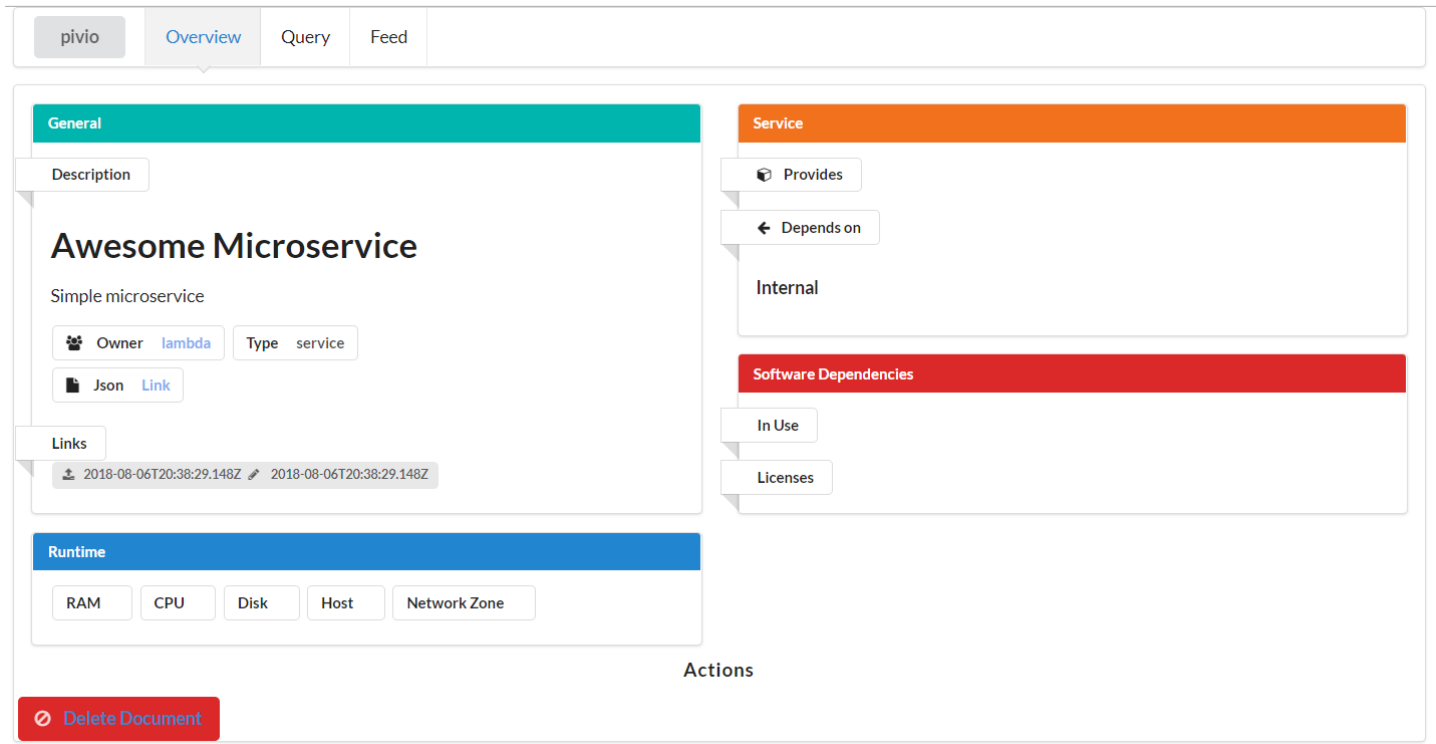
1. Integration of JIRA as a business specific information source
2. Integration the EA Tool (iteraplan)
3. Implementation of crawler jobs (Cloudfoundry and Openshift)
4. Alignment on Monitoring-KPIs
5. Implementation of additional visualizations (if needed)
6. Extend data model



Live Demo

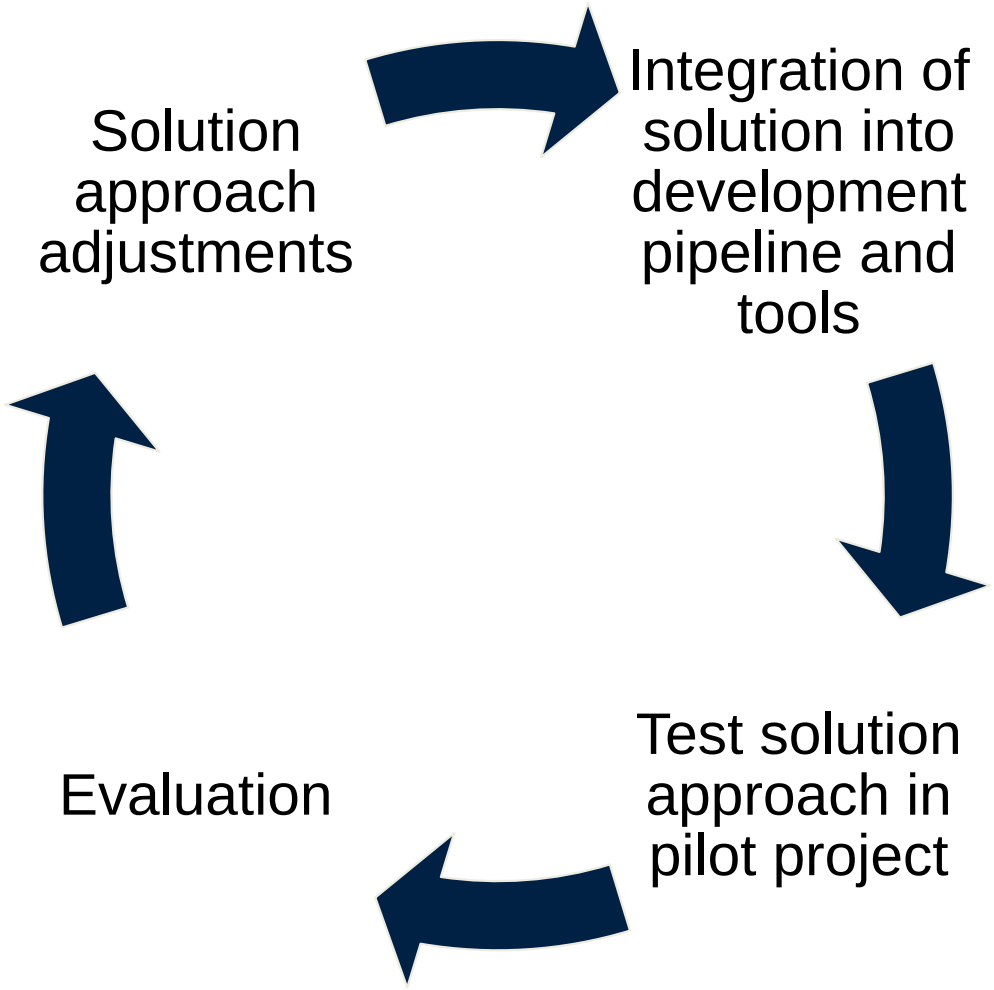

Sources
Build
Get Jira Information
Deploy
Push Documentation
3s
38s
182ms
2min 4s
553ms
3s
38s
182ms
2min 4s
553ms


pivio
Overview
Query
Feed
Quick Search...
Matching Artifacts (16)
Awesome Microservice
lambda
Simple microservice
vor 1 Monat vor 1 Monat
Demokick
Miriam
Simple microservice
vor 2 Tagen vor 2 Tagen
Microservice 1
Nicolas
bla bla bla
ms1
vor 2 Wochen vor 2 Wochen
Microservice 2
Nicolas
bla bla bla
ms2
vor 2 Wochen vor 2 Wochen


pivio
Overview
Query
Feed
General
Description
Awesome Microservice
Simple microservice
Owner lambda
Type service
Json Link
Links
2018-08-06T20:38:29.148Z 2018-08-06T20:38:29.148Z
Service
Provides
Depends on
Internal
Software Dependencies
In Use
Licenses
Runtime
RAM
CPU
Disk
Host
Network Zone
Actions
Delete Document

Backup

Evaluation process



- Solution approach will be implemented in an insurance company
- Tools used during the development:
  - Project management: Jira
  - Continuous Delivery: Jenkins
  - EA Tool: iteraplan
  - Monitoring: Pivio 2.0
- Cloud environment: CloudFoundry and Openshift

Monitoring KPIs

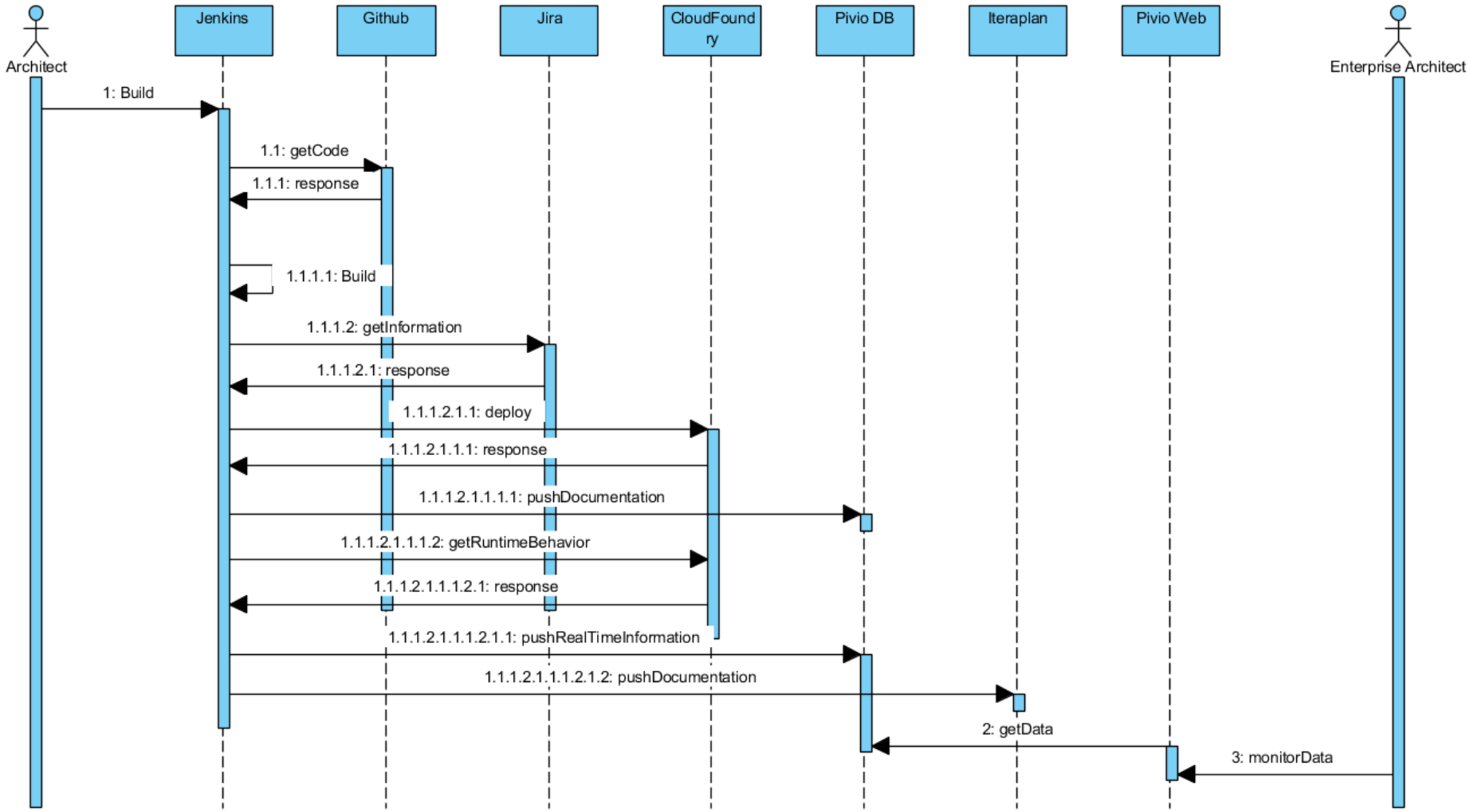
- Cloud clustering as additional visualization
- Backlog-items KPI: How many items are missing. Application portfolio purposes
- Status of applications: running, stopped or crashed.
- Number of deployments per time unit: Which applications change frequently?
- Traffic KPI: Decommission purposes
- Traffic heatmaps: Which applications are important. Relevant for planification and costs. Ratio costs maintenance and costs
- LOC: Maintenance vs Complexity (related to maintenance costs)
- Additional KPIs

Groovy script

```
def callPost(String urlString, String queryString) {
    def url = new URL(urlString)
    def connection = url.openConnection()
    connection.setRequestMethod("POST")
    connection.doInput = true
    connection.doOutput = true
    connection.setRequestProperty("content-type", "application/json;charset=UTF-8")

    def writer = new OutputStreamWriter(connection.outputStream)
    writer.write(queryString.toString())
    writer.flush()
    writer.close()
    connection.connect()

    new groovy.json.JsonSlurper().parseText(connection.content.text)
}
node {
    deleteDir()
    stage('Sources') {
        checkout([
                $class           : 'GitSCM',
                branches         : [[name: "refs/heads/master"]],
                extensions       : [[$class: 'CleanBeforeCheckout', localBranch: "master"]],
                userRemoteConfigs: [[
                                        credentialsId: 'cbf178fa-56ee-4394-b782-36eb8932ac64',
                                        url          : "https://github.com/Nicocovi/MS-Repo"
                                    ]]
                ])
    }
    dir("") {
        stage("Build"){
            sh "gradle build"
        }
stage("Get Jira Information"){
            //TODO
        }
```

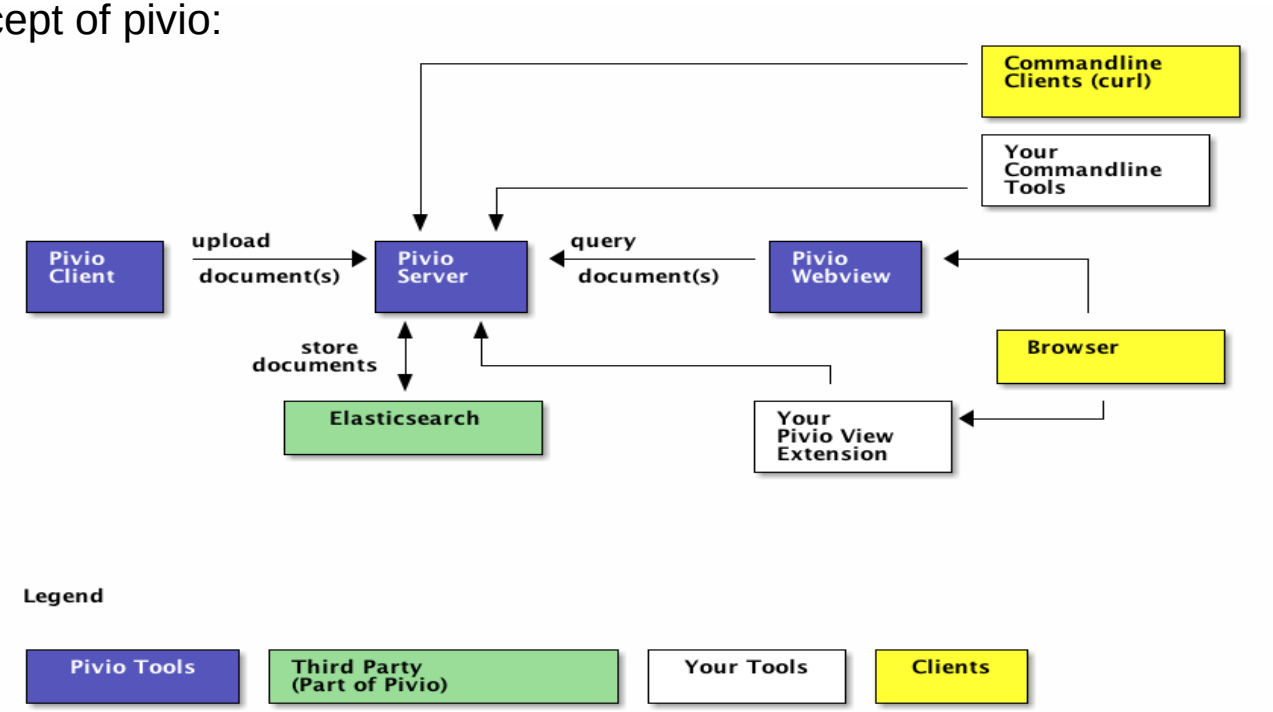
```
stage('Deploy') {
            def branch = ['master']
            def name = "sping-microservice1"
            def path = "build/libs/gs-spring-boot-0.1.0.jar"
            def manifest = "manifest.yml"

              if (manifest == null) {
               throw new RuntimeException('Could not map branch ' + master + ' to a manifest file')
              }
              withCredentials([[
                                  $class          : 'UsernamePasswordMultiBinding',
                                  credentialsId   : '98c5d653-dbdc-4b52-81ba-50c2ac04e4f1',
                                  usernameVariable: 'CF_USERNAME',
                                  passwordVariable: 'CF_PASSWORD'
                              ]]) {
               sh 'cf login -a https://api.run.pivotal.io -u $CF_USERNAME -p $CF_PASSWORD --skip-ssl-validation'
               sh 'cf target -o ga72hib-org -s masterarbeit'
               sh 'cf push sping-microservice1 -f '+manifest+' --hostname '+name+' -p '+path
            }
        }
        stage("Push Documentation"){
            try {
                callPost("http://192.168.99.100:9123/document", "{\"id\": \"0987654321\", \"name\": \"Kick-off-App\",
\"owner\": \"Nico\", \"description\": \"bla\", \"short_name\": \"serviceAZ12\", \"type\": \"service\"}") //Include protocol
            } catch(e) {
                // if no try and catch: jenkins prints an error "no content-type" but post request succeeds
            }
        }//stage
    }
}
```

4. Solution: Sequence diagram



Pivio.io

- What is Pivio ?
  Pivio is a service registry for humans.
- Why Pivio ?
  - Overview for platforms, especially for microservice environment.
  - Reusability of services
  - A growing number of services means also a challenge not only for developers.
  - Which service runs where? What does it do? Who is responsible for that?
- Concept of pivio:



Pivio data model

Pivio needs certain mandatory fields:

- **id:** Unique id in pivio.
- **name:** The name of the artefact.
- **short_name:** A very brief name for the service.
- **type:** The type of this artefact. Values could be **service**, **library** or **mobile_app**.
- **owner:** Which team is responsible for this artefact.
- **description:** What does this service do?