

# A prototypical tool to discover architecture changes based on multiple monitoring data sources for a distributed system

Patrick Schäfer, 22.05.2017, Munich  
Advisor: Martin Kleehaus

Chair of Software Engineering for Business Information Systems (sebis)  
Faculty of Informatics  
Technische Universität München  
[www.matthes.in.tum.de](http://www.matthes.in.tum.de)

## Motivation

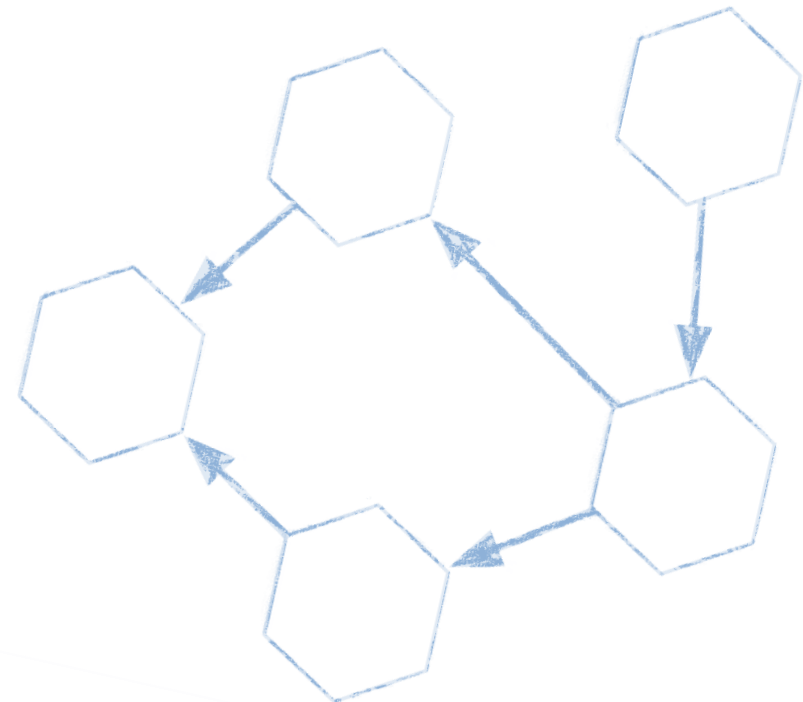
- Modeled & Runtime Architecture
- Problem Statement

## Research Questions

## Approach

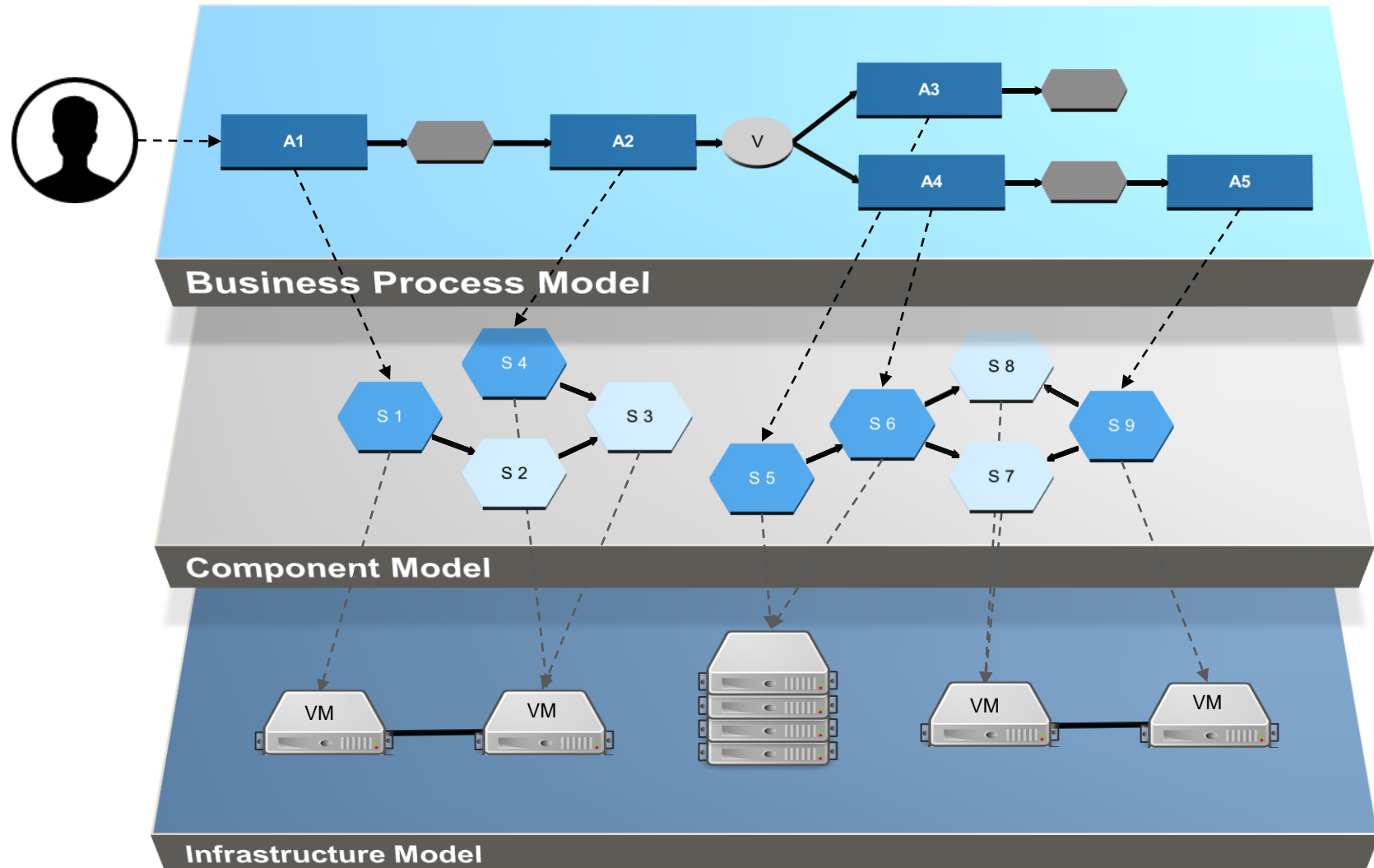
- Architecture Discovery Idea
- Discovery System Environment
- User Interface Mockups
- Limitations

## Timeline



# Motivation

## Modeled Enterprise Architecture

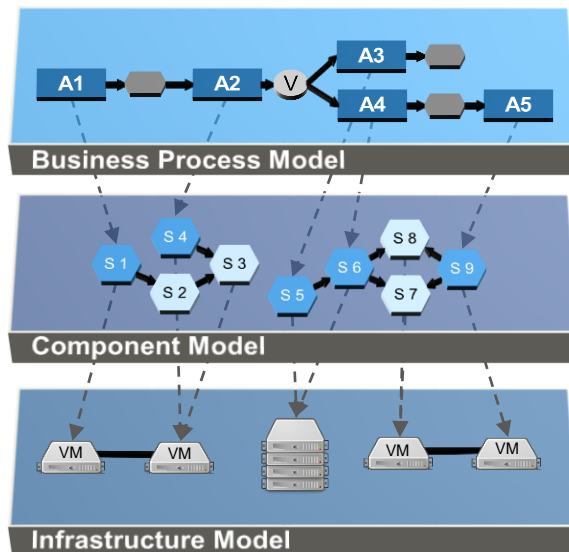


### Necessity of knowledge about runtime architecture

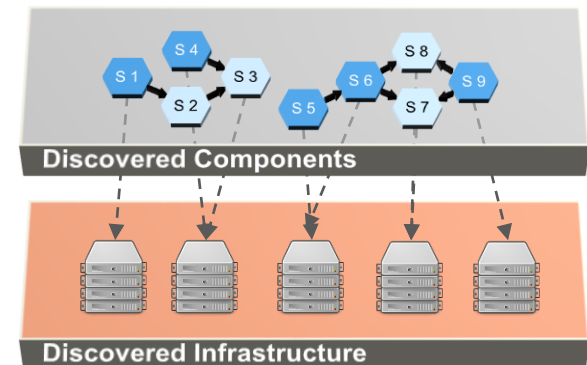
- Detecting deviations from modeled architecture
- Compliance violations / SLA violations
- Semi-automated root cause analysis
- Semi-automated Impact analysis

### State of the art solution

#### Application Runtime Architecture Discovery



Architecture discovery by distributed tracing



**Problem: Current discovery solutions don't own knowledge about the entire architecture including dependencies between and within the architectural layers**

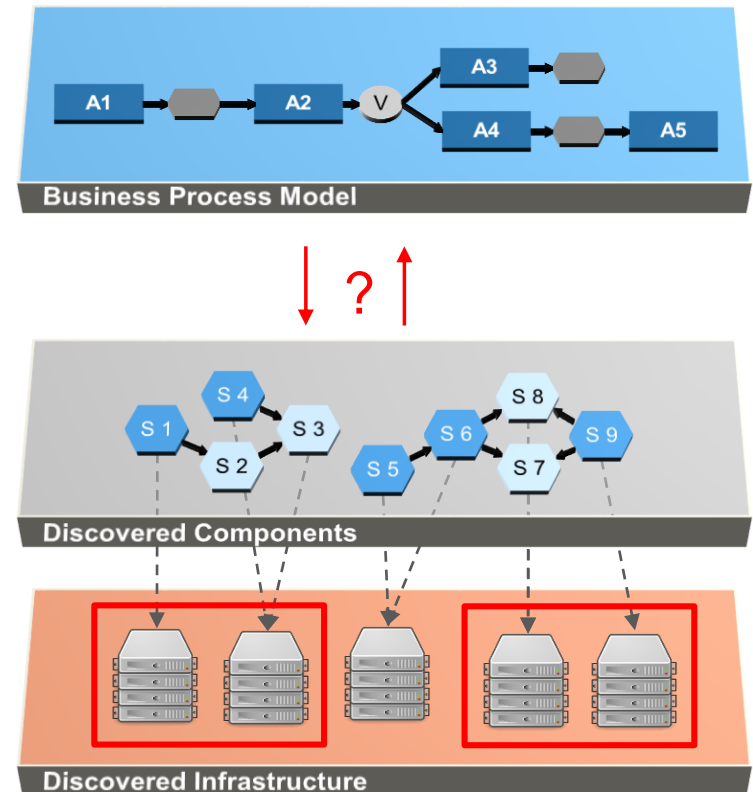
- Interlayer dependencies between components and business activities unknown
- Intralayer dependencies on infrastructure layer unknown

### Consequences

- comparison of modeled and runtime architecture not possible across all layers
- root cause analysis is challenging and time consuming
- Incomplete impact analysis

### Objective

- a system, that owns knowledge about all realtime architecture components as well as their inter- and intralayer dependencies

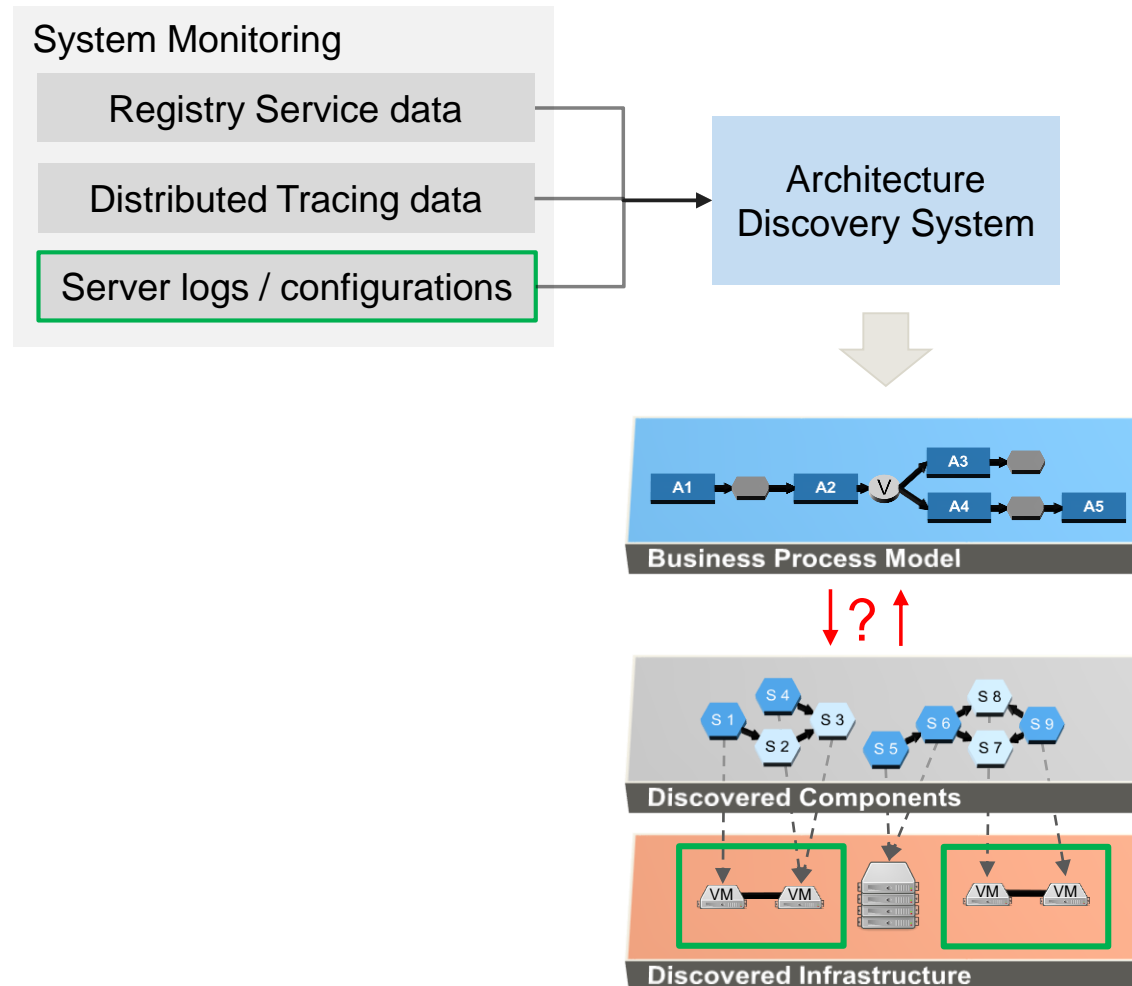


- 1 How to **discover the Microservice Architecture** by applying distributed tracing?
- 2 What are **component relationship types** and how to discover them automatically?
- 3 How to discover **concurrency and synchronization**?
- 4 How to **recognize changes** in the Microservice Architecture?
- 5 How to **categorize services** regarding their target purpose?
- 6 How to provide a **smart user interface** for adding **business semantic** on top of business services?

# Approach

## Architecture Discovery Idea

### Continuous architecture discovery by utilizing multiple data sources



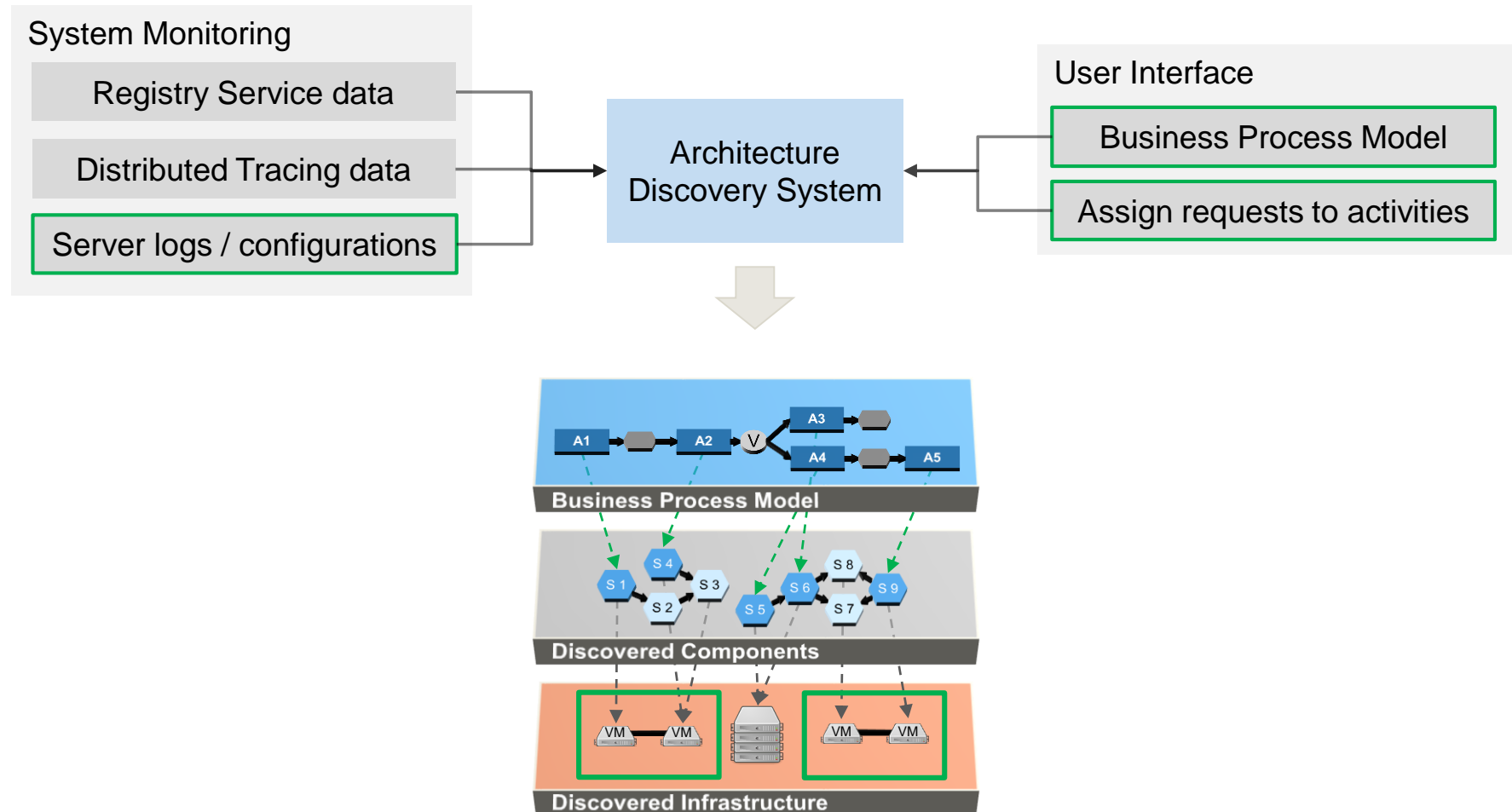
# Approach

## Architecture Discovery Idea

**Continuous architecture discovery**  
by utilizing multiple data sources

+

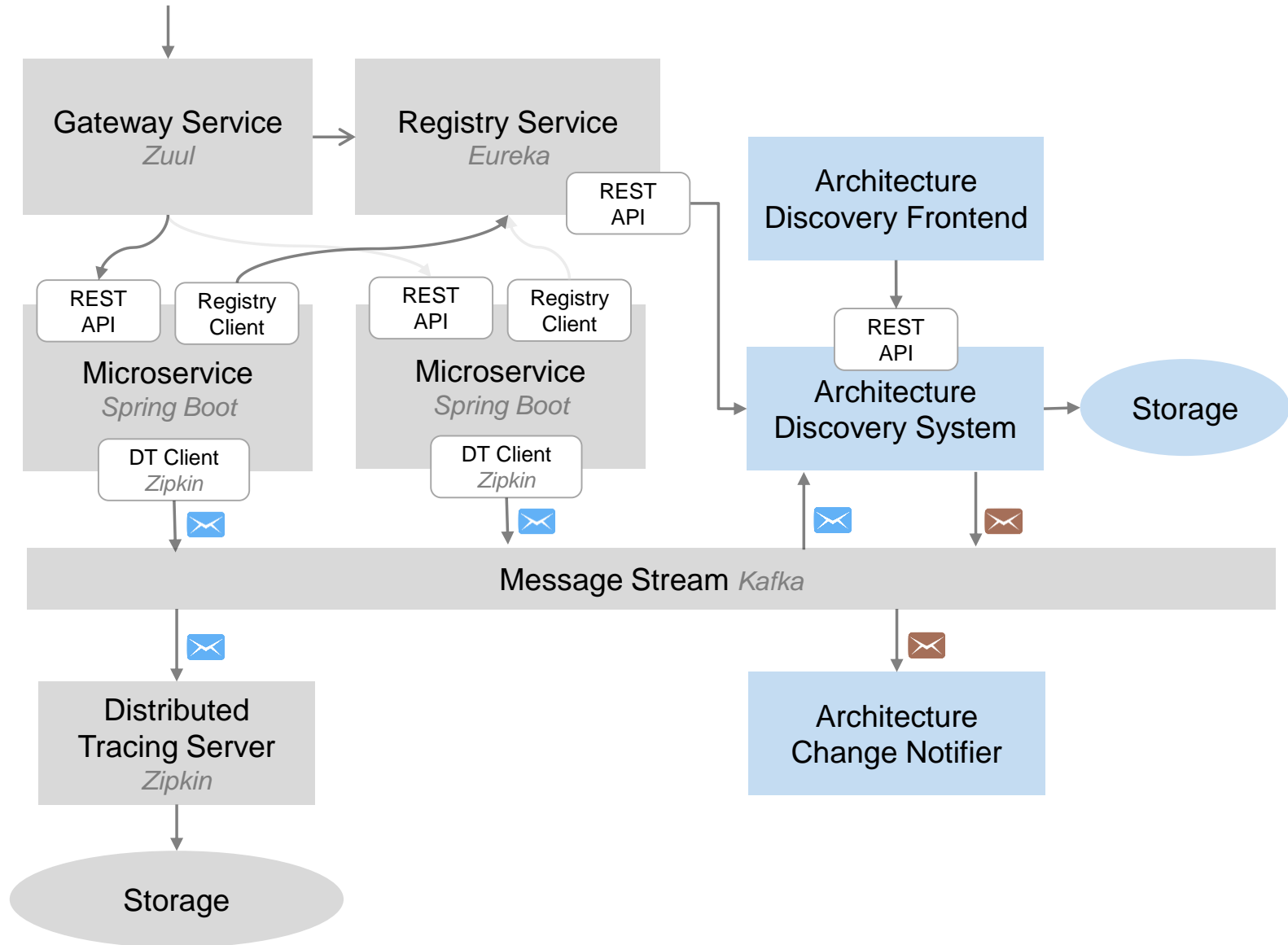
**Business semantic**  
by semi-automated way





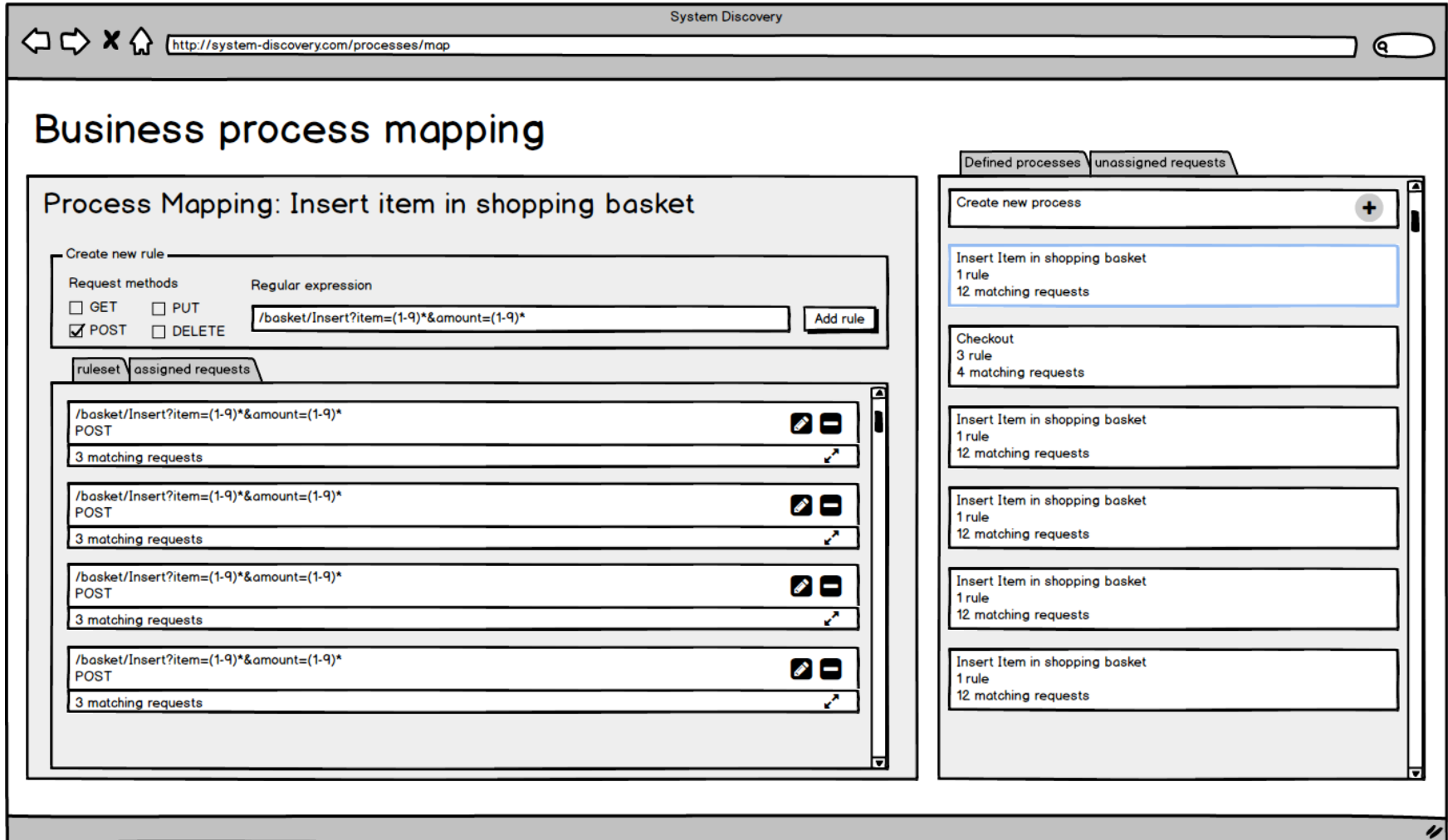
# Approach

## Discovery System Environment



# Approach

## User Interface Mockup



# Approach

## User Interface Mockup

System Discovery

http://system-discovery.com/processes/map

### Dependency Structure Matrix

Description	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Business Process 1	1	-		X	X			X	X		X	X				
Business Process 2	2	X	-	X	X	X	X	X	X	X	X	X				
Business Activity 1.1	3			-				X	X		X	X				
Business Activity 1.2	4				-				X							
Business Activity 2.1	5					-				X						
Business Activity 2.2	6						-		X	X						
Webservice 1	7							-	X			X				
Webservice 2	8								-		X					
Webservice 3	9									-		X				
Server 1	10										-					
Server 2	11											-				

**Time of Architecture representation**

**Filtering**

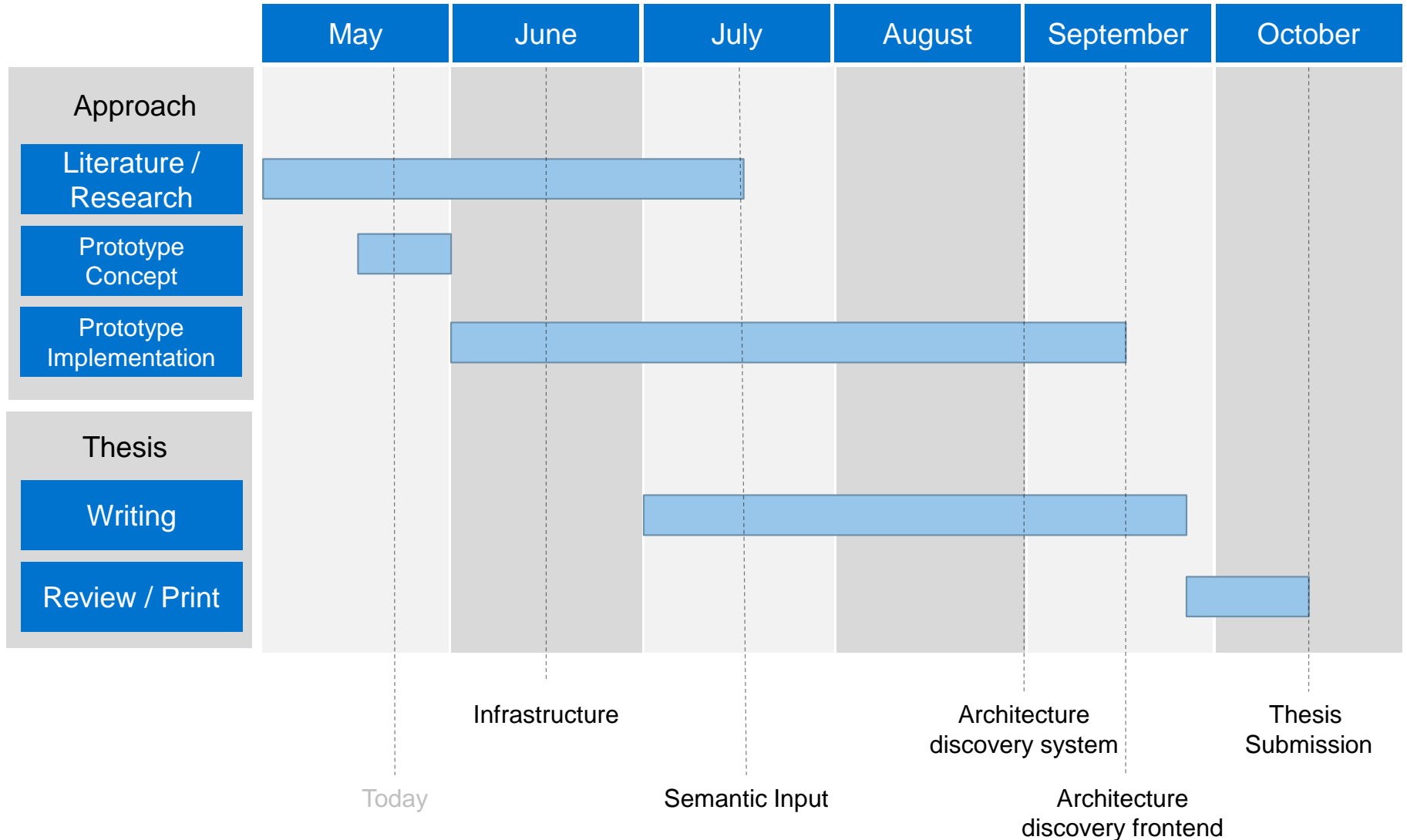
**Business process**

**Levels**

Processes       Processactivities

Components       Infrastructure

# Timeline



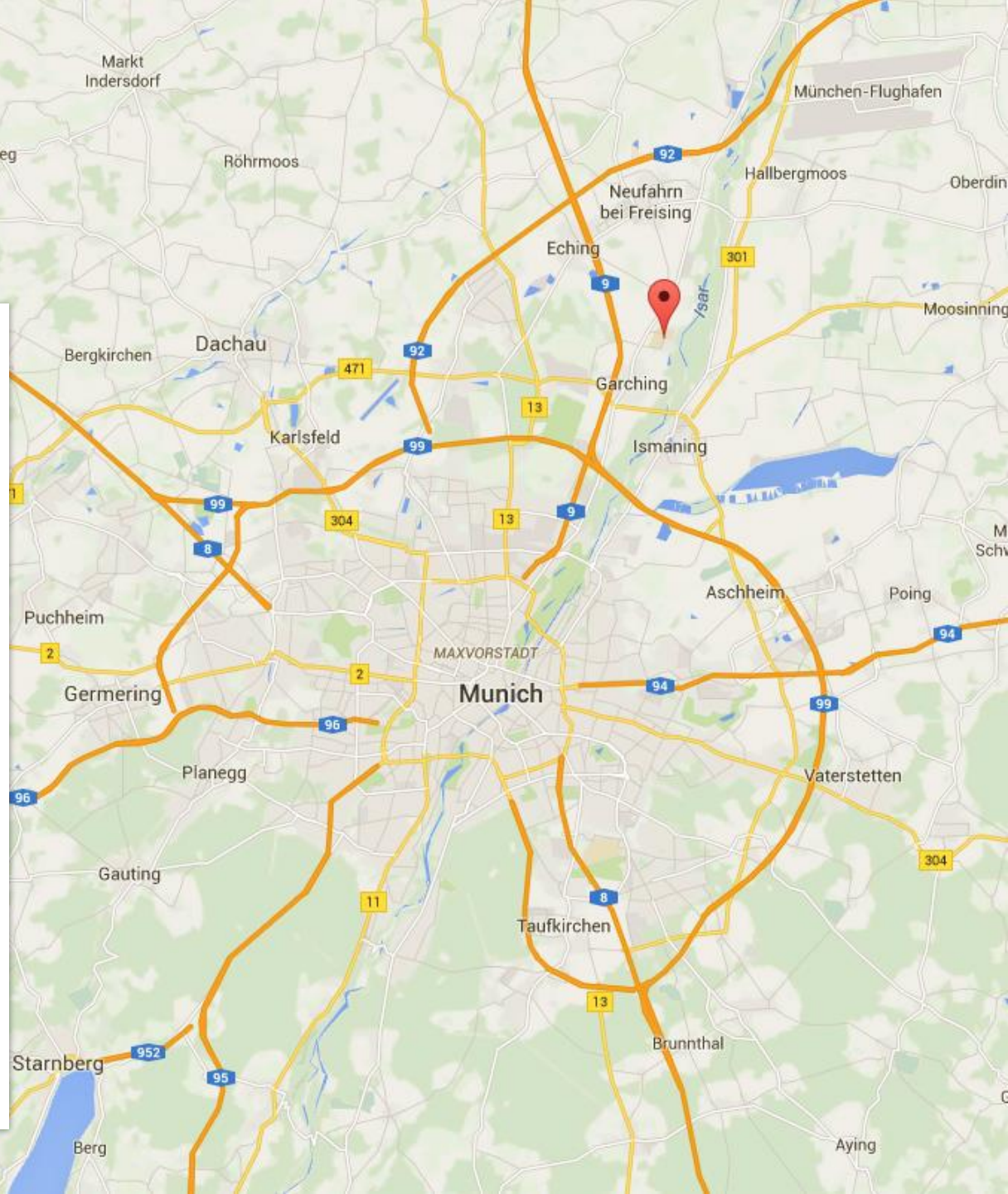


B.Sc.  
**Patrick Schäfer**

Technische Universität München  
Faculty of Informatics  
Chair of Software Engineering for  
Business Information Systems

Boltzmannstraße 3  
85748 Garching bei München

patrick.schaefer@tum.de  
[www.matthes.in.tum.de](http://www.matthes.in.tum.de)



Backup



# Approach

## Resulting datastructure

Dependency Structure Matrix (t)

Dependency Structure Matrix (t+1)

Dependency Structure Matrix (t+2)

Description	#	1	2	3	4	5	6	7	8	9	10	11
Bu Bu Business Process 1	1	-		X	X							
Bu Bu Business Process 2	2	X	-	X	X	X	X					
Bu Bu Business Activity 1.1	3			-								
Bu Bu Business Activity 1.2	4				-							
W Bu Business Activity 2.1	5					-						
W W Business Activity 2.2	6						-					
W W Webservice 1	7							-				
S W Webservice 2	8								-			
S S Webservice 3	9									-		
S S Server 1	10										-	
S S Server 2	11											-

### Foundation for automated

- detection of architecture changes
- comparison of modeled and runtime architecture
- root cause analysis
- impact analysis

# Approach

Discovery data source: Server logs / configurations

