

IOWA STATE UNIVERSITY



Architecture Recovery

Lotfi ben Othmane

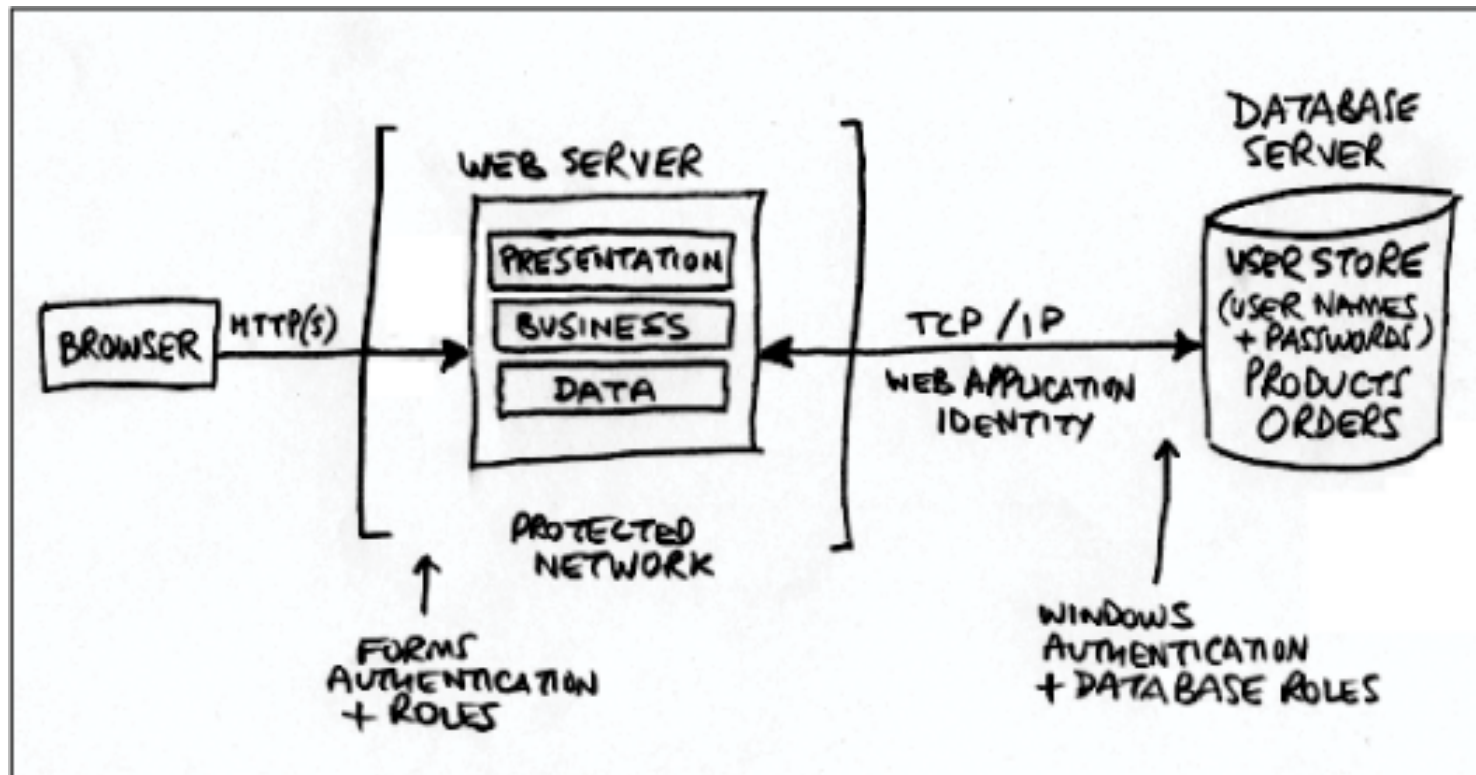
Spring 2020

Architecture Recovery

- What is it?
- Why do we need it?
- How to do it?

Prescriptive Architecture

Prescriptive architecture describes the **expected** architecture of software (often designed one)



Descriptive Architecture

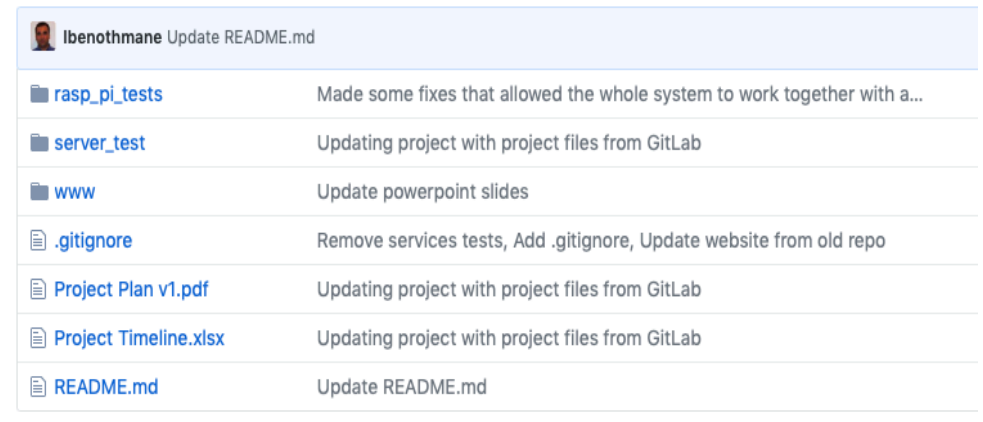
Descriptive architecture is the **as-implemented** architecture of software

ibnothmane Update README.md	
📁 rasp_pi_tests	Made some fixes that allowed the whole system to work together with a...
📁 server_test	Updating project with project files from GitLab
📁 www	Update powerpoint slides
📄 .gitignore	Remove services tests, Add .gitignore, Update website from old repo
📄 Project Plan v1.pdf	Updating project with project files from GitLab
📄 Project Timeline.xlsx	Updating project with project files from GitLab
📄 README.md	Update README.md

Descriptive Architecture

How would you get the **descriptive architecture (as-implemented)** of a software

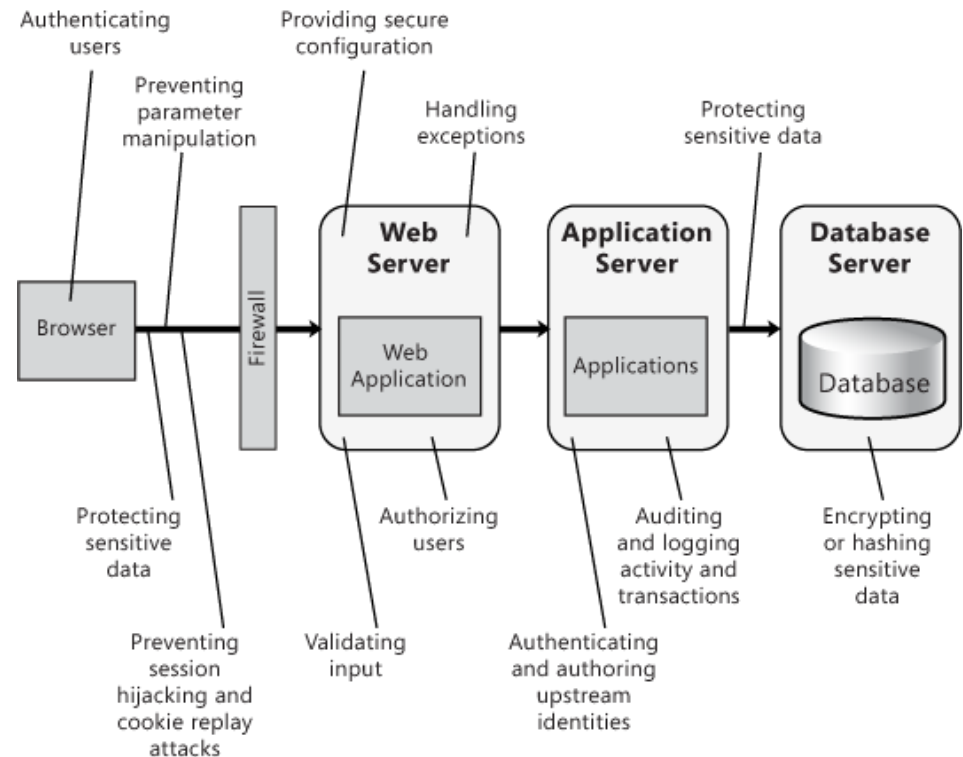
1. Inspect the code
2. See the execution
3. Communicate with customers
4. Read the documentation
5. Other techniques



Ibenothmane Update README.md	
📁 rasp_pi_tests	Made some fixes that allowed the whole system to work together with a...
📁 server_test	Updating project with project files from GitLab
📁 www	Update powerpoint slides
📄 .gitignore	Remove services tests, Add .gitignore, Update website from old repo
📄 Project Plan v1.pdf	Updating project with project files from GitLab
📄 Project Timeline.xlsx	Updating project with project files from GitLab
📄 README.md	Update README.md

Architecture and Software Evolution

- Software are developed in iterations and versions
- Software architecture describes a given iteration/version of software



Architecture Gap

- **Prescriptive architecture** describes the expected architecture of software (often designed one)
- **Descriptive architecture** is the as-implemented architecture of software
- Descriptive architecture and prescriptive architecture are **often different**

Impacts of the Architecture Gap

- Architectural decisions made based on the perspective architecture may be wrong
- Mechanisms for quality attributes could be broken
- The assumption that the software satisfies a set of quality attributes may be wrong

How to address the gap between prescriptive architecture and descriptive architecture?

Solutions for the Architecture Gap

Alternatives:

1. Update the architecture document and redevelop the architecture diagrams
2. Recover the architecture of the software from the code

Descriptive Architecture

- **Ground truth** – the architecture of a software that has been verified as accurate by the architects.
- **Architecture recovery** – is the extraction and analysis of a software architecture

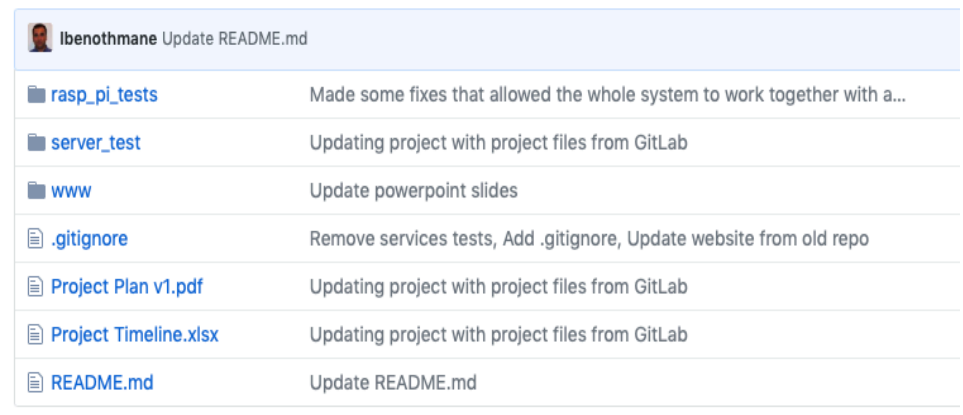
Architecture Recovery

- **Architecture recovery** is the extraction and analysis of a software architecture
- Current tools cluster the software code into packages
- Recovery techniques often are based on the **call graph** of the software

Architecture Recovery Approaches

- Simple methods – Use the folders' structure
- Text-based methods – Mine the text code
- Dependency-based methods – Use function calls network

This is not an exhaustive list



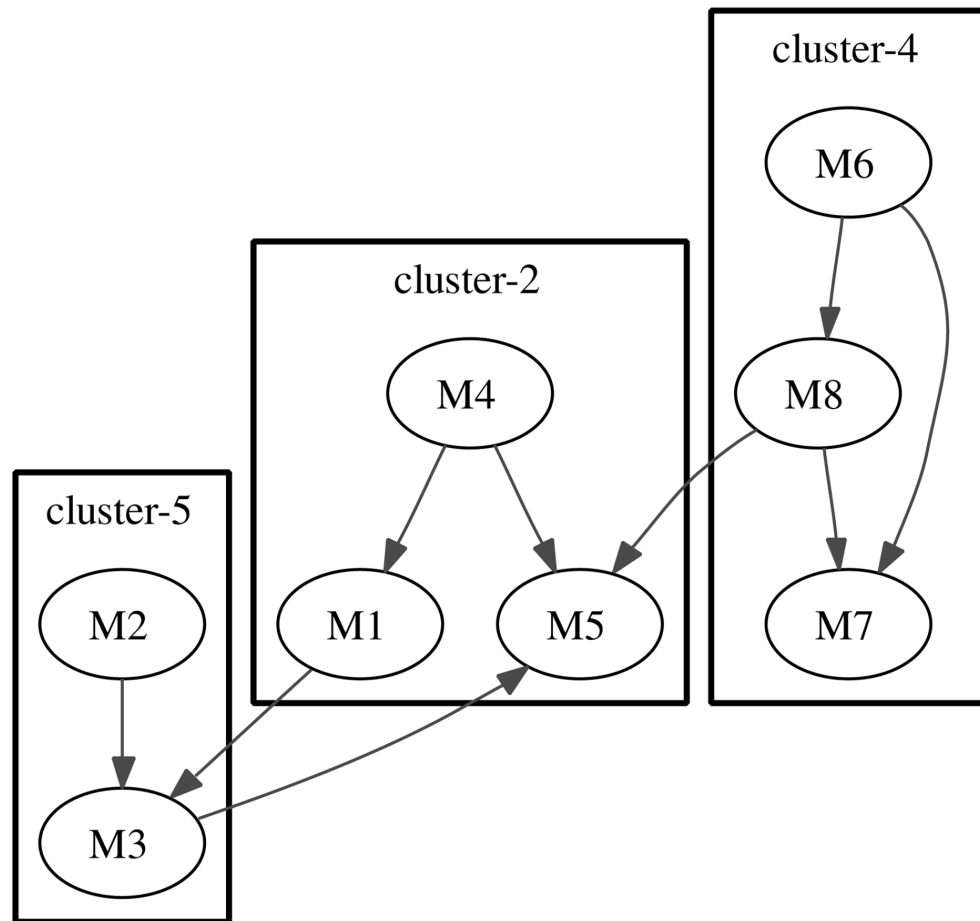
File/Folder	Description
rasp_pi_tests	Made some fixes that allowed the whole system to work together with a...
server_test	Updating project with project files from GitLab
www	Update powerpoint slides
.gitignore	Remove services tests, Add .gitignore, Update website from old repo
Project Plan v1.pdf	Updating project with project files from GitLab
Project Timeline.xlsx	Updating project with project files from GitLab
README.md	Update README.md

Call Graph

- **Call graph** represents the calling relationships between nodes. Each node represents a function (or module) and each edge (f, g) represents calls of function (or module) f to function (or module) g .
- Use code analysis tools to extract the CFG.

Graph Clustering

- Architecture recovery becomes a clustering problem
 - Each method defines its own clustering feature



Architecture Recovery Methods

1. Algorithm for comprehension-driven clustering – cluster code based on dominance
2. Weighted combined algorithm – cluster based on a distance metric between the clusters
3. LIMBO – Uses a metric named distributed cluster feature
4. Architecture recovery using concerns – uses information retrieval and machine learning techniques
5. Zone base recovery – based on natural language semantics of identifiers found in the code
6. Bunch – cluster code using modularization metric

<https://dl.acm.org/citation.cfm?id=2819022>

Bunch Metric

- **Assumes:** well-designed software systems are organized into cohesive subsystems that are loosely interconnected.
- **Interconnectivity** - dependencies between the modules of two distinct subsystems
- **Intra-connectivity** - dependencies between the modules of the same subsystem
- **Modularization Quality** – trade-off between Interconnectivity and Intra-connectivity

Bunch Metric

- **Intra-connectivity** - Coefficient of number of edges in the cluster to potential number of NODES in the cluster

$$A_i = \frac{\mu_i}{N_i^2}$$

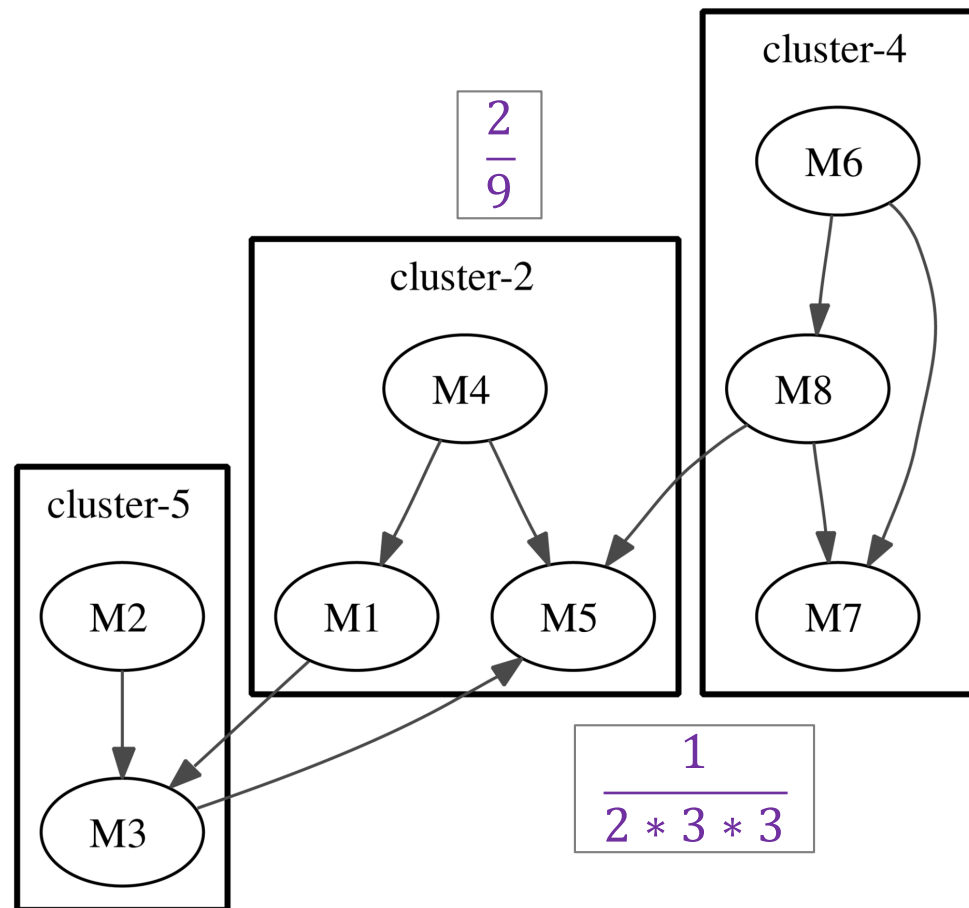
- **Interconnectivity** - Coefficient of number of edges between cluster i and cluster j to double the number of nodes of cluster i multiply by the number of nodes of cluster j . (0 if in the same cluster.)

$$E_{i,j} = \frac{\varepsilon_{i,j}}{2 \times N_i \times N_j}$$

- **Modularization Quality** -
$$\begin{cases} \frac{\sum_{i=1}^k A_i}{k} - \frac{\sum_{i,j=1}^k E_{i,j}}{\frac{k(k-1)}{2}} \\ A_1 & (k = 1) \end{cases}$$

Bunch Metric

It uses hill-climbing and genetic algorithms to solve the optimization problem



Compute the:

1. Intra-connectivity
2. Inter-connectivity

Of the module of the component: Data-Collector

https://github.com/lbenothmane/FleetManagement/tree/master/rasp_pi_tests

WALA CFG

T.J. Watson Libraries for Analysis (WALA)

The screenshot shows the GitHub repository page for WALA. The browser address bar displays "GitHub, Inc. [US] | https://github.com/wala/WALA". The repository name "wala / WALA" is shown with 21 watches, 146 stars, and 87 forks. Navigation tabs include Code, Issues (53), Pull requests (3), Projects (0), Wiki, and Insights. The repository description is "T.J. Watson Libraries for Analysis" with a link to "http://wala.sourceforge.net". Language tags for "static-analysis", "java", "javascript", and "android" are present. Statistics show 5,343 commits, 3 branches, 11 releases, and 25 contributors. Action buttons include "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download". A recent commit by "juliandolby" is highlighted, with the message "Merge pull request #265 from liblit/java-8-build-configuration" and "Latest commit 5ac8bf8 2 days ago". Below this, a commit from "com.ibm.wala-repository" is shown with the message "Rename \"...-feature\" to \"..._feature\" in subdirs and features" and "3 months ago".

WALA CFG

```
digraph "DirectedGraph" {
graph [concentrate = true];center=true;fontsize=6;node [
color=blue,shape="box"fontsize=6,fontcolor=black,fontname=Arial];edge [
color=black,fontsize=6,fontcolor=black,fontname=Arial];
  "Node: synthetic < Primordial, Lcom/ibm/wala/FakeRootClass, fakeRootMethod()V > Context:
  Everywhere" [ label="Node: synthetic < Primordial, Lcom/ibm/wala/FakeRootClass, fakeRootMethod()
  > Context: Everywhere"]
  "Node: synthetic < Primordial, Lcom/ibm/wala/FakeRootClass, fakeWorldClinit()V > Context:
  Everywhere" [ label="Node: synthetic < Primordial, Lcom/ibm/wala/FakeRootClass, fakeWorldClinit()
  > Context: Everywhere"]
  "Node: < Application, Lorg/compiere/util/SecureEngine, main([Ljava/lang/String;)V > Context:
  Everywhere" [ label="Node: < Application, Lorg/compiere/util/SecureEngine, main([Ljava/lang/String;
  )V > Context: Everywhere"]
  "Node: < Application, Lorg/compiere/util/SecureEngine, <clinit>()V > Context: Everywhere" [
  label="Node: < Application, Lorg/compiere/util/SecureEngine, <clinit>()V > Context: Everywhere"]
  "Node: < Application, Lorg/compiere/util/Util, main([Ljava/lang/String;)V > Context: Everywhere" [
  label="Node: < Application, Lorg/compiere/util/Util, main([Ljava/lang/String;)V > Context:
  Everywhere"]
  "Node: < Application, Lorg/compiere/util/Util, <clinit>()V > Context: Everywhere" [ label="Node: <
  Application, Lorg/compiere/util/Util, <clinit>()V > Context: Everywhere"]
  "Node: < Application, Lorg/compiere/plaf/CompierePLAF, main([Ljava/lang/String;)V > Context:
  Everywhere" [ label="Node: < Application, Lorg/compiere/plaf/CompierePLAF, main([Ljava/lang/String;
  )V > Context: Everywhere"]
  "Node: < Application, Lorg/compiere/plaf/CompierePLAF, <clinit>()V > Context: Everywhere" [
  label="Node: < Application, Lorg/compiere/plaf/CompierePLAF, <clinit>()V > Context: Everywhere"]
  "Node: < Application, Lorg/compiere/util/Language, main([Ljava/lang/String;)V > Context:
  Everywhere" [ label="Node: < Application, Lorg/compiere/util/Language, main([Ljava/lang/String;)
  > Context: Everywhere"]
  "Node: < Application, Lorg/compiere/util/Language, <clinit>()V > Context: Everywhere" [
  label="Node: < Application, Lorg/compiere/util/Language, <clinit>()V > Context: Everywhere"]
}
```

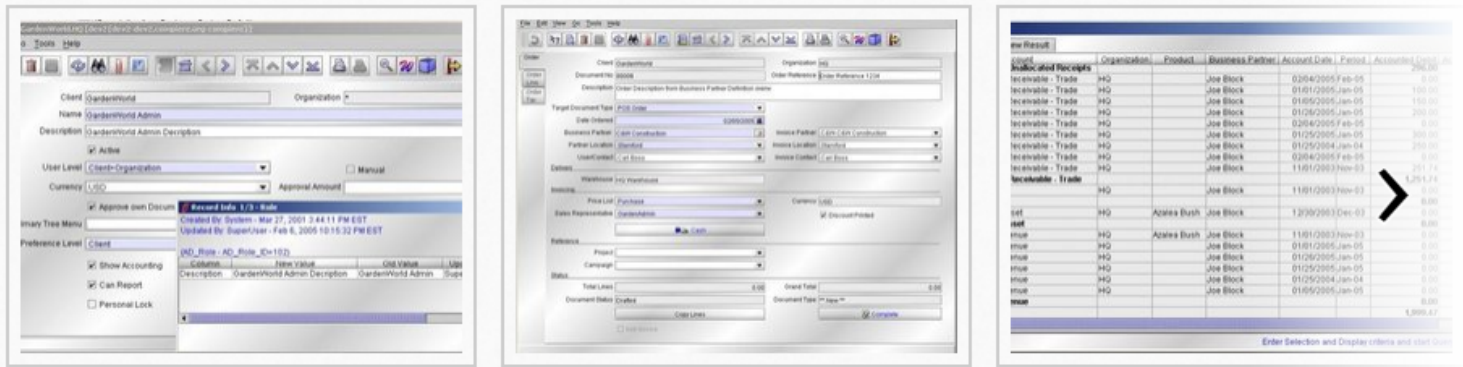
Case Study - Compiere

Overview

Compiere ERP+CRM is the leading open source ERP solution for...

[Read More](#)

[Compiere ERP + CRM Business Solution Web Site »](#)



<https://sourceforge.net/projects/compiere/>

Case Study - Compiere

Suppose that you are asked to develop a new feature for Compiere, how would you proceed?

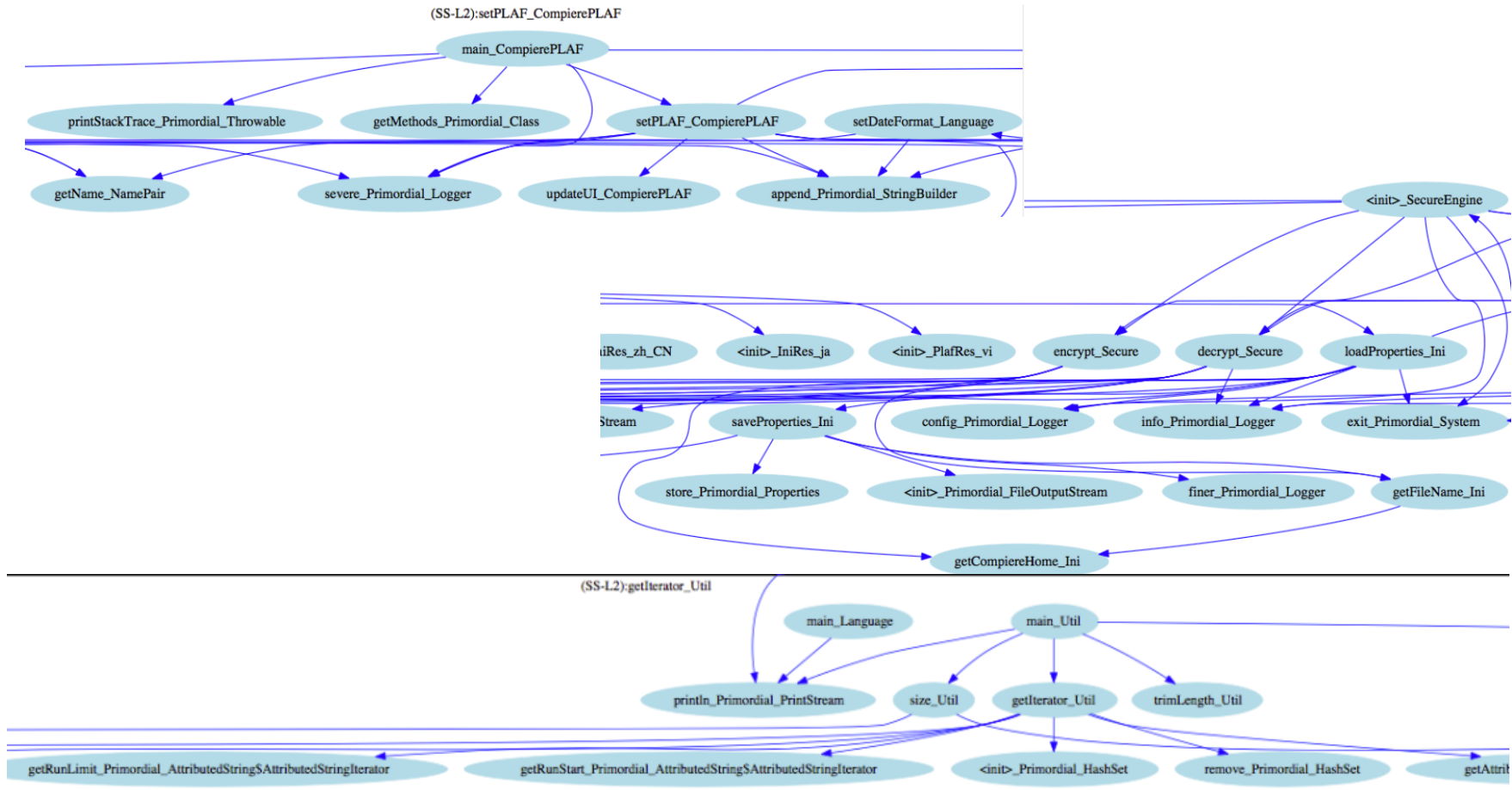
1. Read the source code
2. Give it to an expert with the software
3. Read the documentation
4. Say you cannot do it
5. Recover the architecture of the software

Case Study - Compiere

How would you recover the architecture of Compiere?

1. Use the folders structure
2. Use a cluster-based approach
3. Use a text-based approach

Case Study – Compiere - Clusters



Example of Challenges

1. Code analysis tools add fake nodes.
2. Code includes dependency – e.g., Java-based code includes JRE java methods.
3. Methods may have similar names but in different modules – may confuse code analysis.
4. The number of clusters is huge, e.g., 2000

Case Study – Compiere - Clusters

1. **Util** - main class for utilities
2. **New Instance** - when user logs in, creates new instance with respective look & feel and language
3. **Logger** - generation and storage of logs
4. **Secure Engine** - responsible for implementing security policies within the application and initializing security
5. **NameValuePair** - stores/retrieves/modifies user related data
6. **List Resources** - lists user resources on login
7. **Main_CompierePLAF** - provides look & feel
8. **GetLanguage** - retrieves the language for user
9. **Encrypt_SecureEngine** - provides data encryption capabilities
10. **Hashing** - stores hashmap of user data

Summary

- **Prescriptive architecture** describes the expected architecture of software (often designed one)
- **Descriptive architecture** is the as-implemented architecture of software
- Descriptive architecture and prescriptive architecture are **often different**
- **Architecture recovery** is the extraction and analysis of a software architecture

Thank you

Questions?