Archicheck overview

A simple solution to architecture degradation

About architecture

About architecture degradation

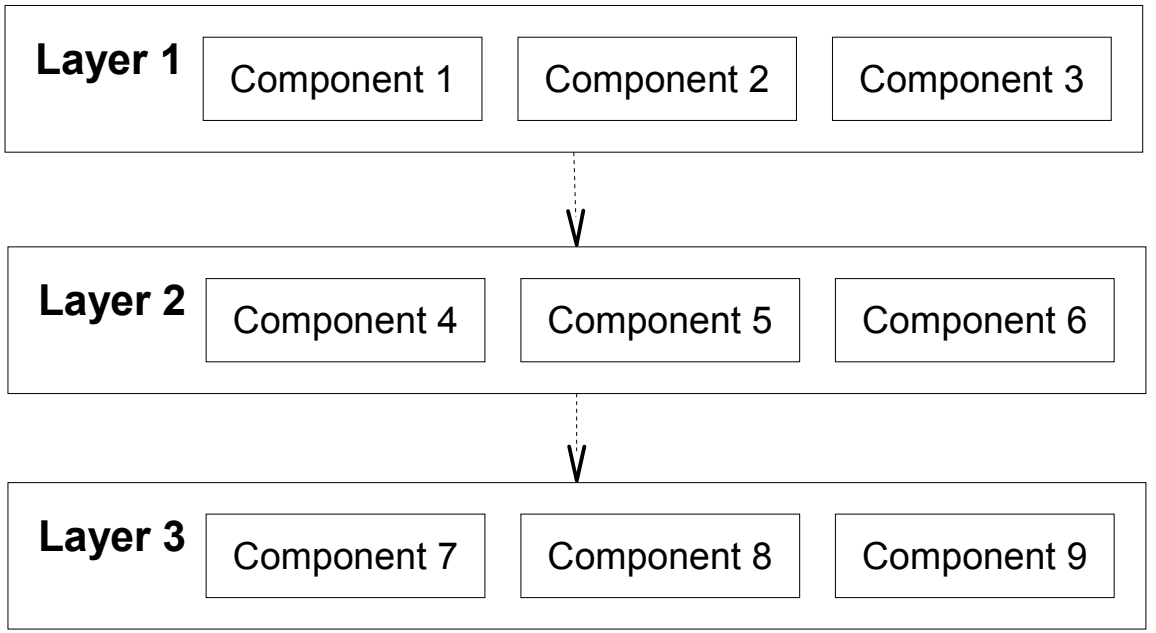
Why does architecture degrade

- Architecture decision are stated :
  - Sometime (partially) in doc
  - Sometime (partially) in models
  - Generally nowhere explicitly

- Even when explicitly stated, most architecture decision are no more checked after initial build

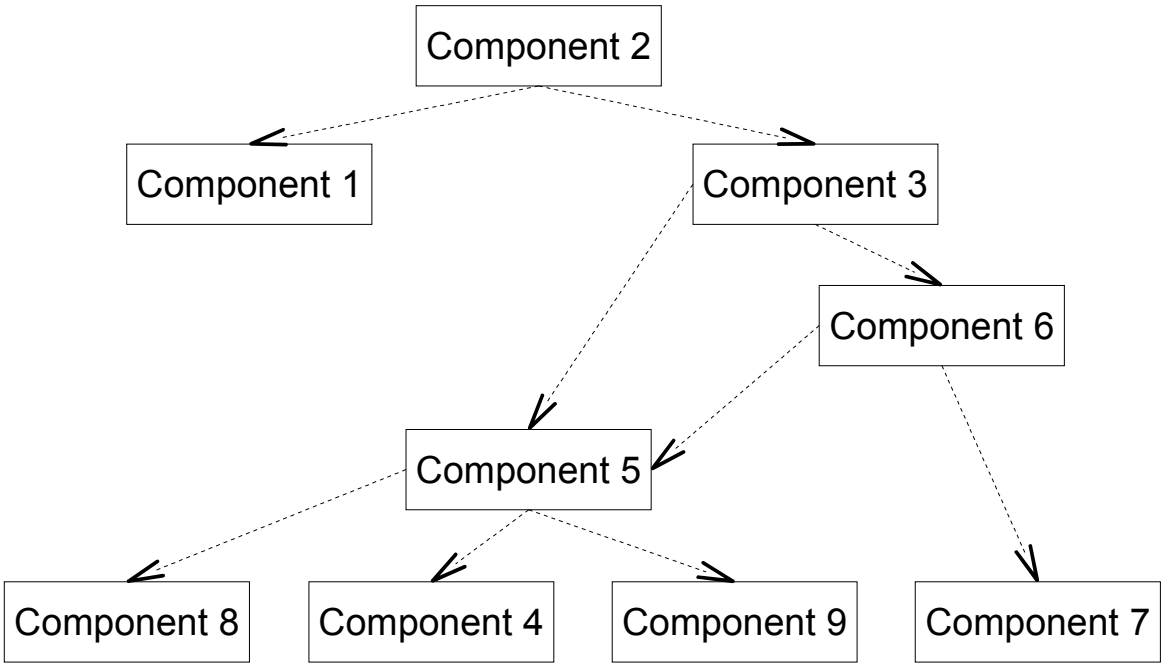
Degradation example 1/4 : The original design

What does the architect build : A simple layered system



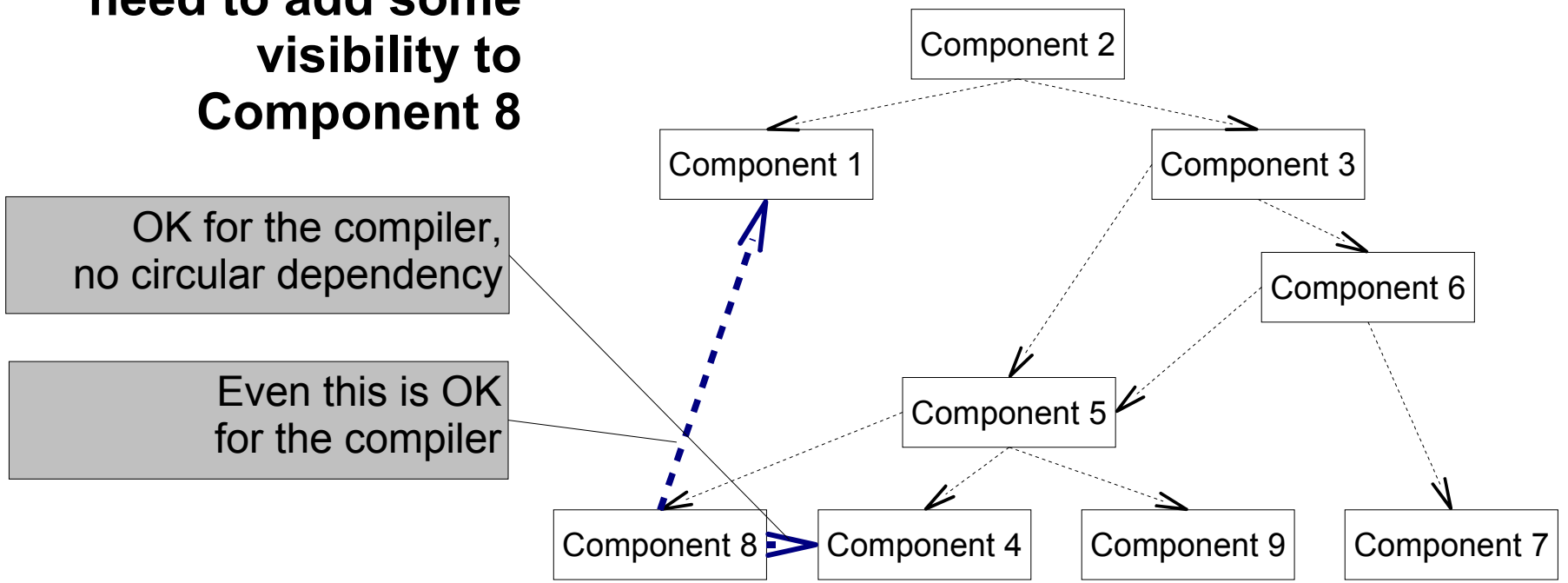
Degradation example 2/4 : What is actually in the code

What shows in the code : compilation units and dependencies



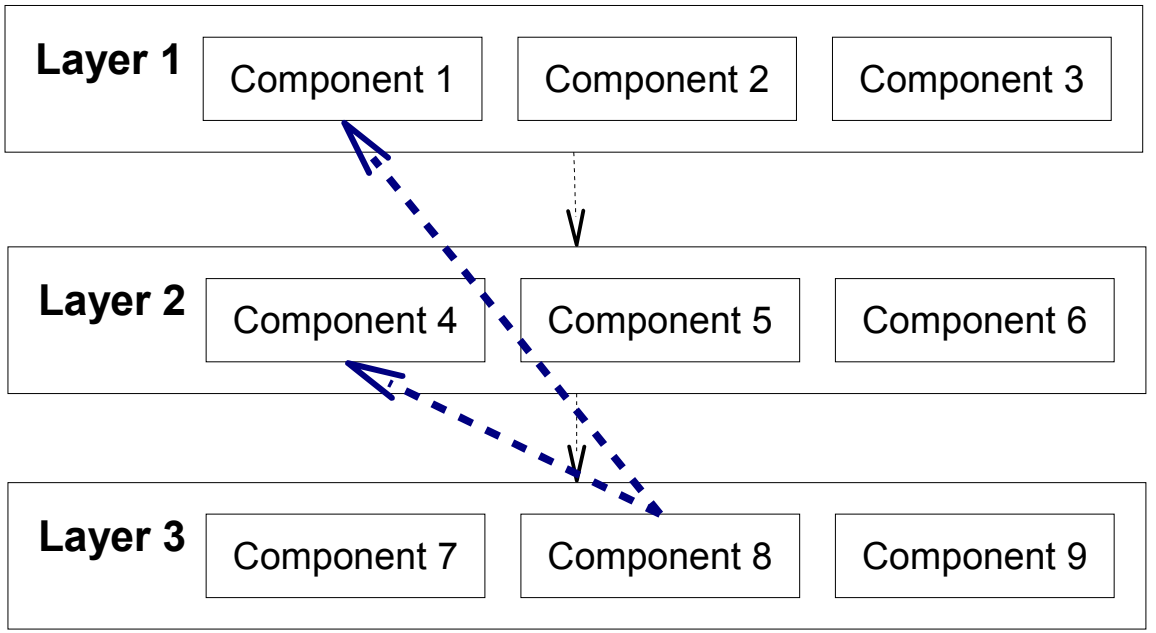
Degradation example 3/4 : What if I add those “with”?

To fix a bug, I need to add some visibility to Component 8



Degradation example 4/4 : But does it fit with the original design?

No!



Solution

- Add the needed semantic
- Where?
  - Design : ADL, modelling tool and UML extension
  - Code : external tool like Archicheck, or directly in future programming languages

Why a code based tools?

- Most of the semantic is already in the code
- There is always code

Archicheck definition

- Archicheck provides a simple and natural language to describe a software architecture in terms of partition and dependencies
- Archicheck is a tool that checks the conformance of sources with one of those description

Archicheck design

Archicheck use

- Write the description with whatever text editor
- Run it in your Makefile (typically in the “make test”)

Use hints

- The description is a text file containing simple english sentences : put it “as is” in your documention
- Building a framework or a library? Provide it with his own archicheck definition, to help using it the right way

Why use Archicheck?

- It's easy to install and learn
- It's free (n'alourdi pas le process) : put it in the Makefile and forget!
- Raise explicit the most important part of your architecture
- Continuously enforces code coherence with your design