

What's coming...

What's the fuzz all about?

What is (good, bad or ugly) architecture?

Mixing the mode

Real life

Getting it done



Reasons for an architecture assessment

- There are problems, rooted in the architecture
 - Performance is degrading
 - Costs for further development are rising
 - Changes take longer and longer
 - The build process is erroneous
- Other objectives might be
 - Understand as-is situation
 - Check results of a provider
 - Prepare hand-over of software to another team
 - Establish enterprise standards
- Each Audit differs in its focus
- Evaluate the architecture not its documentation

Which is the best car...?

1.



3.



2.



4.



....for these roads...?



What is good architecture anyway

An architecture which...

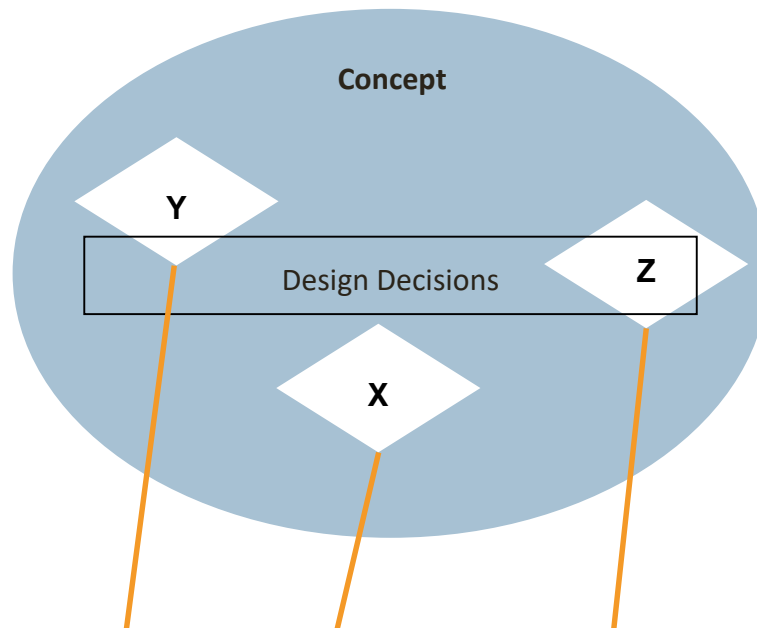
- is comfortable to implement?
- is cheap?
- does show its beauty?
- is easy to understand?
- does enable reuse?
- uses the hottest technology?
- does support testing?
- is compliant with EA guidelines?
- does facilitate change?



Architecture is...

- „... stuff that is hard to change later.“ – Martin Fowler
 - “... the construction plan of a system”. – Gernot Starke
 - “...what describes the coarse grained pieces of an application.” – Neil Ford
 - “...the structure of the system, its elements, the externally visible properties of those elements, and the relationships among them.” – Len Bass et al.
 - “... beneficially relating elements.” – Kent Beck
- ... a set of concepts which satisfy functional and non-functional requirements of a system or an organization.
 - There is no system without architecture. Sadly, sometimes it's created ad-hoc.

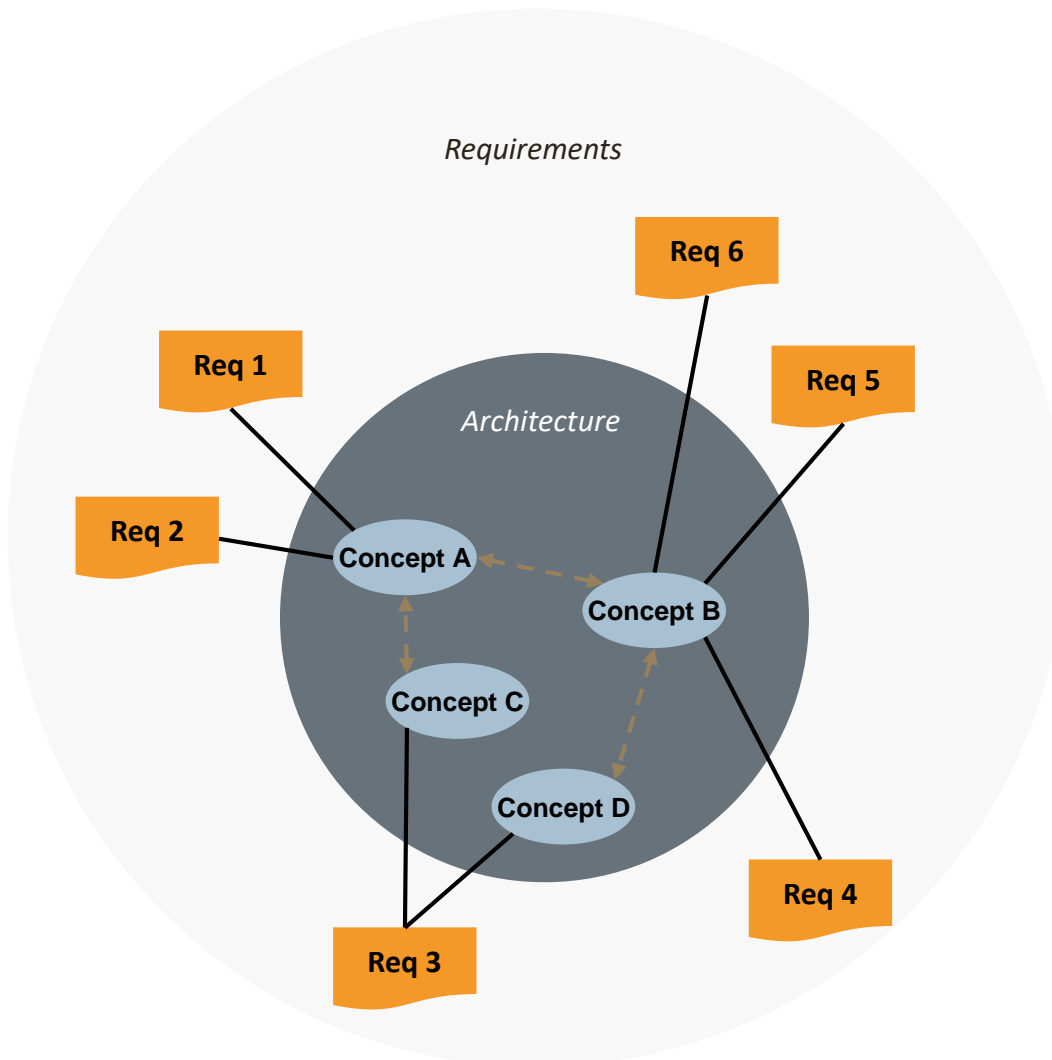
A concept contains one or more design decisions



Example 1: “We utilize **web services** and **messages** to communicate with the backend. **The frontend** does select any one of these depending on business case.”

Example 2: “„We use an in memory cache for master data, which is refreshed after every 24 hours.“

The architecture is made up of concepts

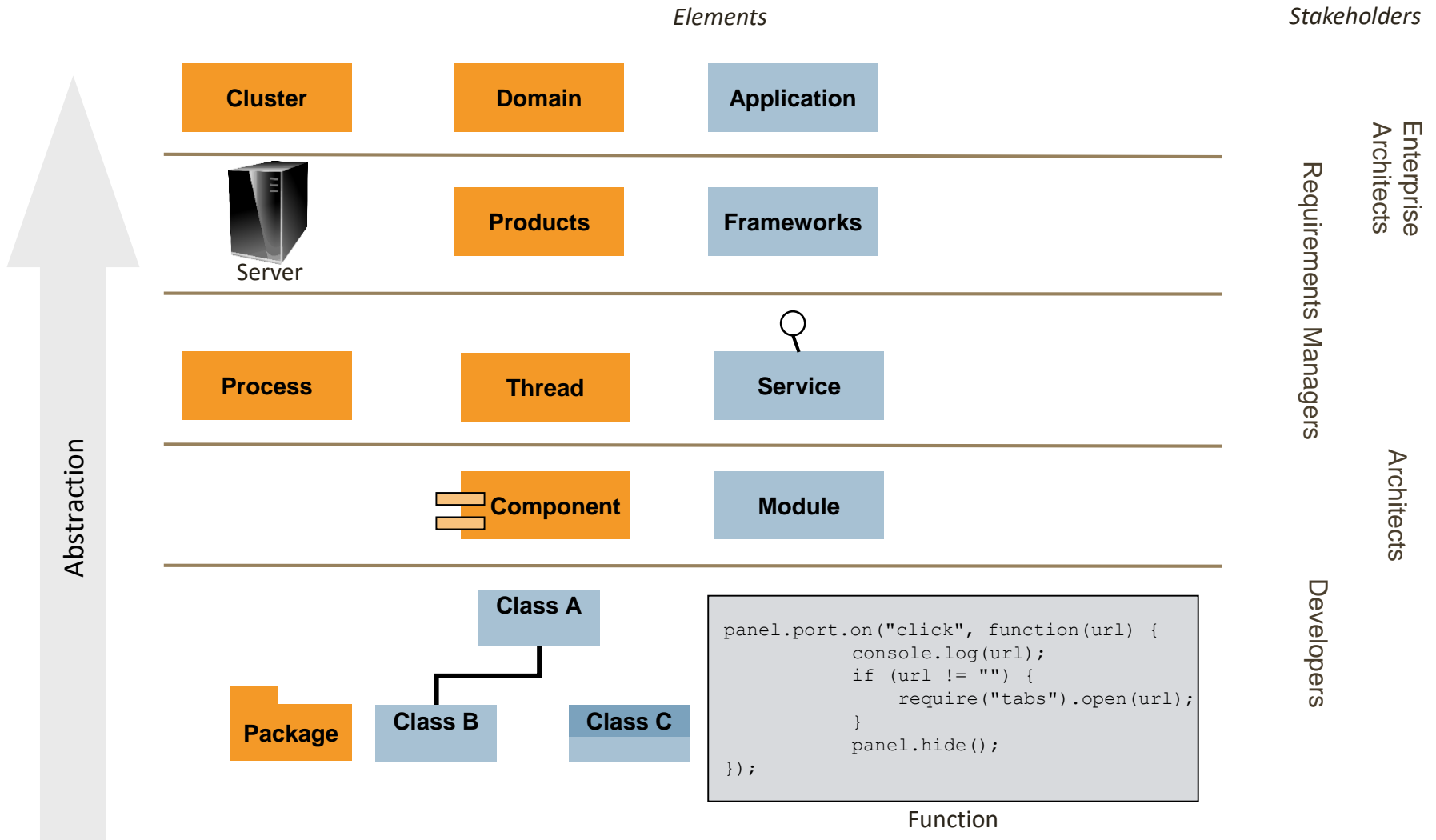


- » Concepts relate to requirements
- » Concepts have dependencies

Concepts use elements

- „We use an in memory cache for master data, which is refreshed after every 24 hours.“
- “Out portlets will be implemented with Vaadin, since it features dynamic carousel/accordion widgets.”
- “We utilize web services and messages to communicate with the backend. The frontend does select any one of these depending on business case.”
- “The mobile version will use the embedded database, since it must be able to handle off-line situations.”
- “The implementation of business services will be separated from the transport layer by being in separate packages and dedicated classes.”
- “We use Bootstrap to implement responsiveness but Sencha Touch for UI components.”
- “All backend communication will go via the ESB, because it enables fail-over.”

Elements are distributed over many tiers

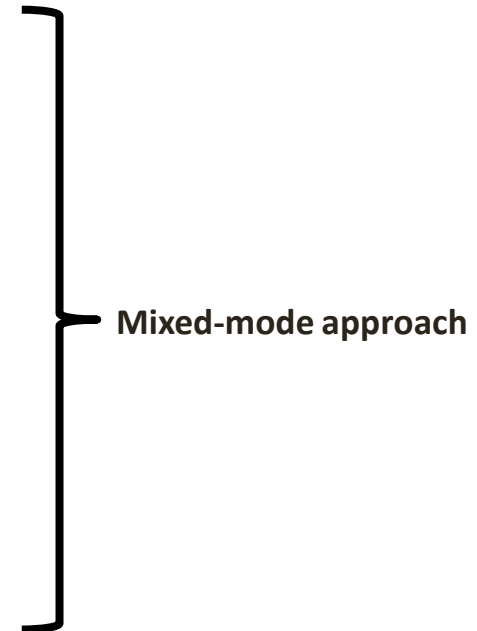


*elements exemplary

Mixed-mode approach is necessary

An architecture assessment must...

- include all (relevant) elements and tiers
- address all stakeholders needed
- include all sources necessary
 - Documents, code, people, systems
- utilizes different methods for each source
 - Interviews
 - Document inspection
 - Scenarios
 - Code review (automatic and manual)
- focus on the essential
 - Core topics, best practice, experience



Quality Scenario – Modular IT architecture

Scenario Name:	New UI device „car on-board display“	
Background:	<p>The architecture must enable modular systems in contrast to silo or black-box systems. The UI must be able to be displayed on new and until today unknown devices and interfaces (given they use a web UI). The overall architecture must enable separated and encapsulated building blocks which can be developed independently and thus can decrease provider dependency.</p>	
Business Goal(s):	Decrease and remove vendor lock-in. Increase quality and improve time-to-market.	
Quality Attributes:	Changeability, Adaptability, Interoperability, Fault Tolerance	
Scenario Components:	Stimulus:	A “car on-board display” has to be supported as touch point for the web shop.
	Stimulus Source:	Market necessities, new requirements
	Artifact (If Known):	(Concrete) UI, UI abstraction layer, digital services layer
	Response / Concepts:	A UI abstraction layer is needed (technology independent). A new “responsive target” has to be developed based on the abstraction layer. There are no direct changes in the abstract UI. With the new responsive target, the UI will adapt itself to the screen size of the on-board unit.
	Response Measure:	Concrete UIs are implemented on top of abstract UI. The new UI is ready to be tested in 6 weeks.
Trade-offs:	Higher complexity in UI layer, performance cost, changes in abstract UI effect all concrete UIs	

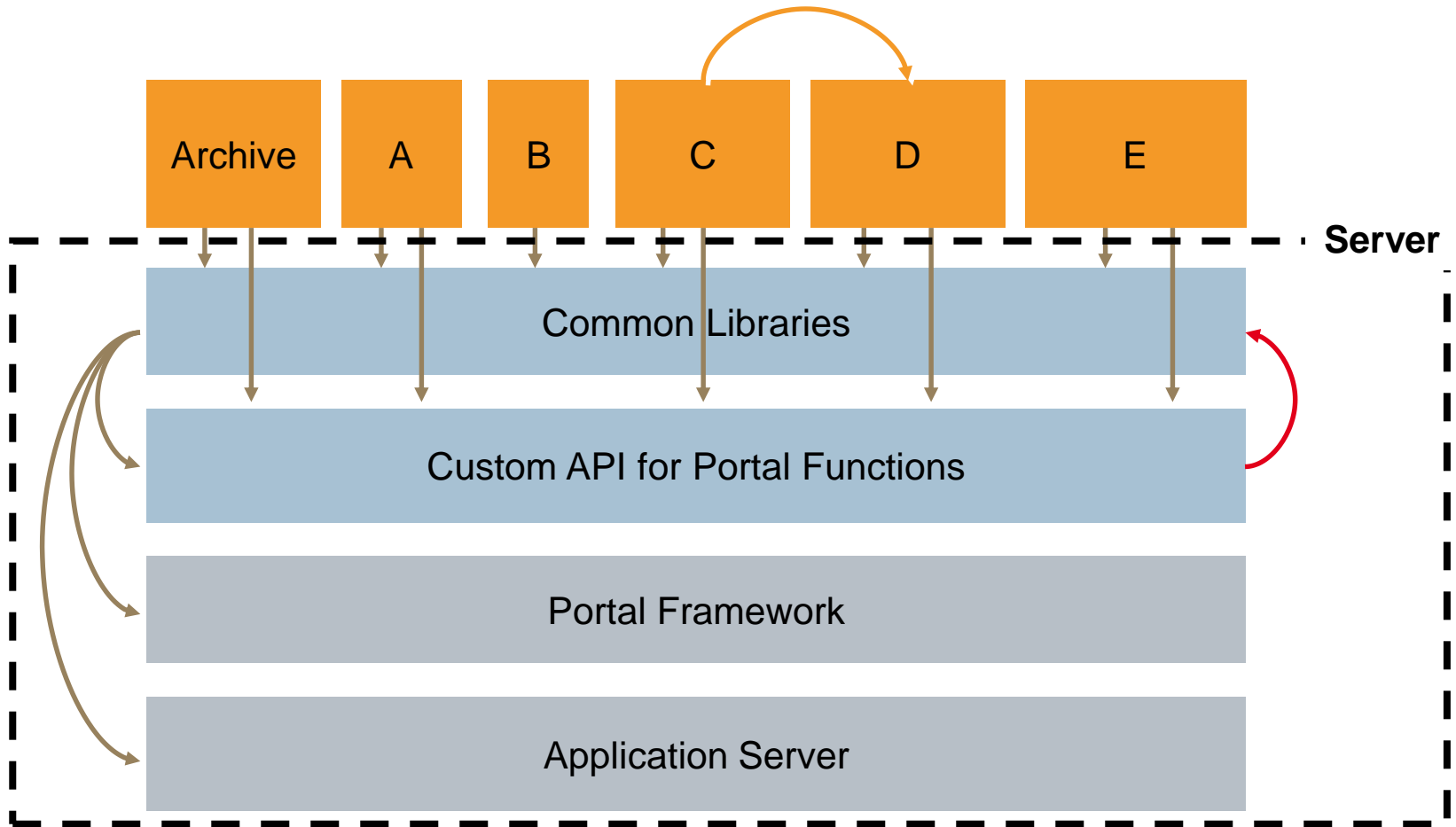
How to assess an architecture

1. Focus: clarify objectives and core questions
2. Find all architectural concepts
 - Utilize mixed-mode approach
3. Validate each concept individually
 - Is it explicit or implicit?
 - Is there minimum 1 requirement per concept?
 - Does the concept satisfy the requirements?
 - Is the concept sensible?
4. Validate concepts together

} Understand the architecture

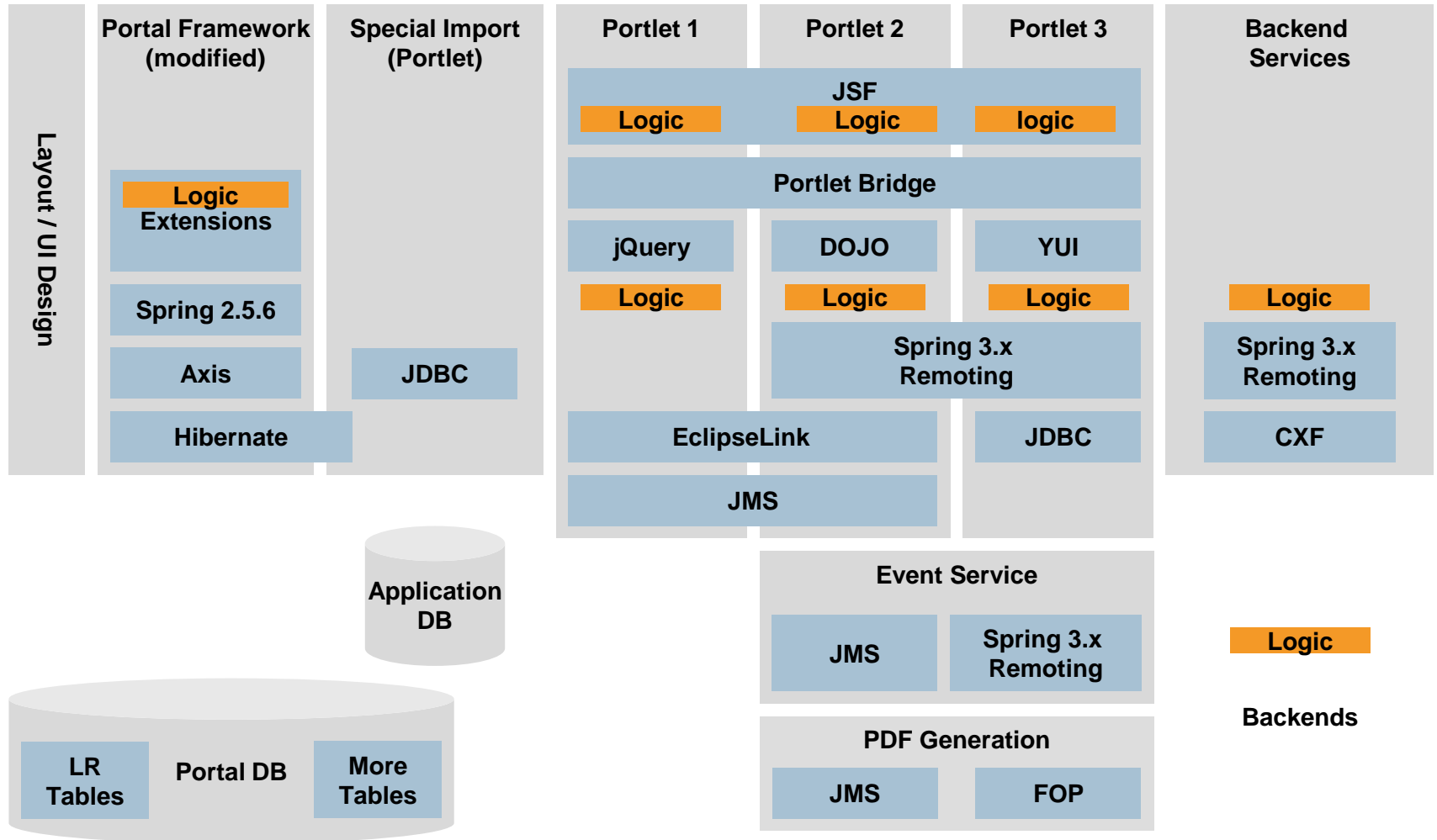
» The quality of the concepts and their fit determine the quality of the architecture.

Example Anti-pattern: Unmanaged dependencies



Found by automated code review

Example Anti-pattern: Distributed business logic



Found by document inspection and interview

Example Anti-pattern: Code Smells

Function Names Should Say What They Do.

```
private String verify(String in) {  
    if (in == null) {  
        return "";  
    }  
    return in;  
}
```



The comments should not describe something that adequately describes itself.

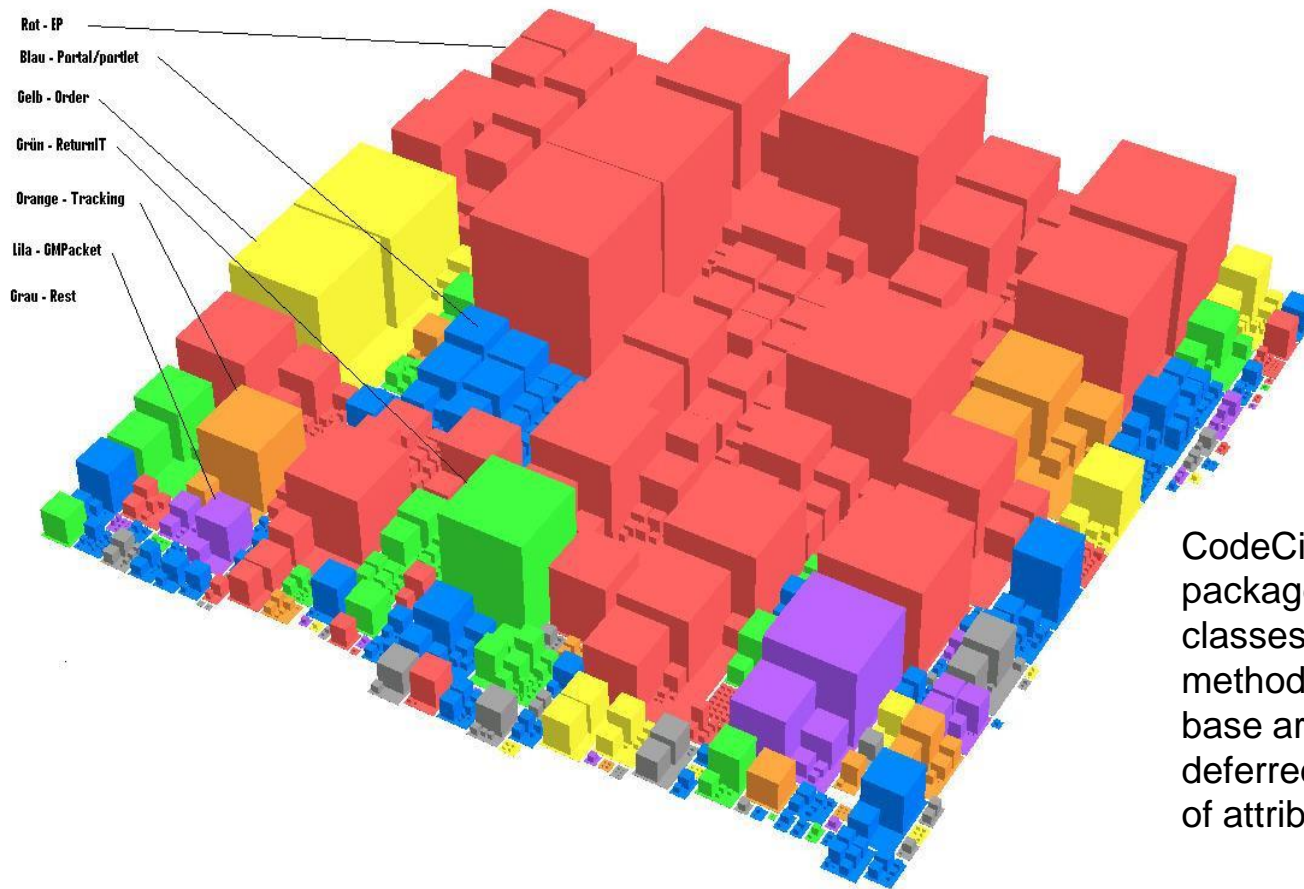
```
try {  
    // Create an appropriate writer according to download type.  
    if (DownloadType.EXCEL_97.equals(downloadType)) {  
        exportWriter = new ExcelExportWriter();  
    } else { // use csv as default  
        exportWriter = new CSVExportWriter();  
    }  
  
    // write the headers  
    exportWriter.writeHeaders(headerLabelList);  
}
```

Functions should have a small number of arguments.

```
String createlabel(long shipperId, Address address, String firstName, String lastName, String companyName, String addressLine1, String addressLine2,  
String street, String houseNumber, String postalCode, String city, String country, String email,  
String weight, String width, String height, String length, String reference1, String reference2) throws CreatelabelException;
```

Found by manual code review

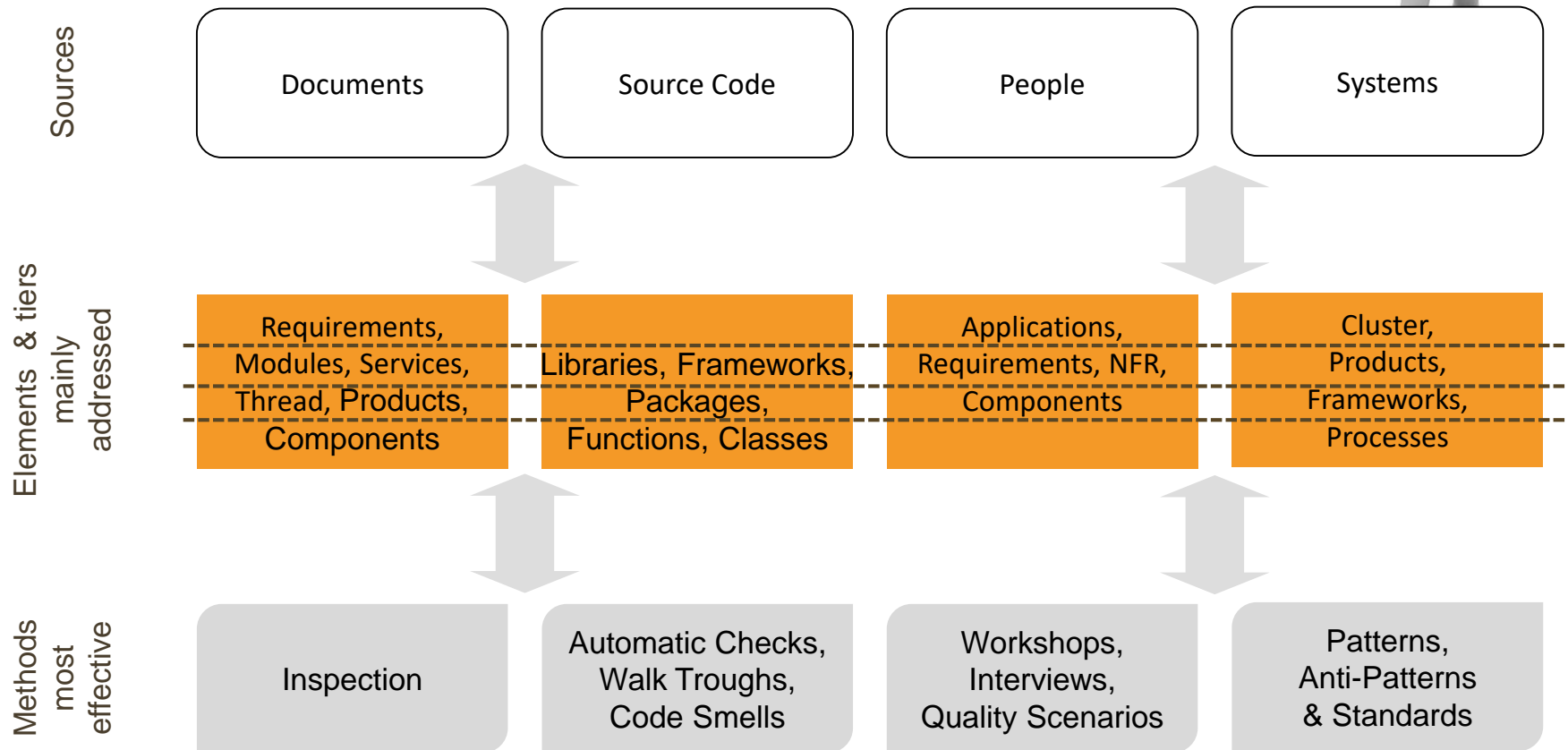
Example Anti-pattern: Classes too complex



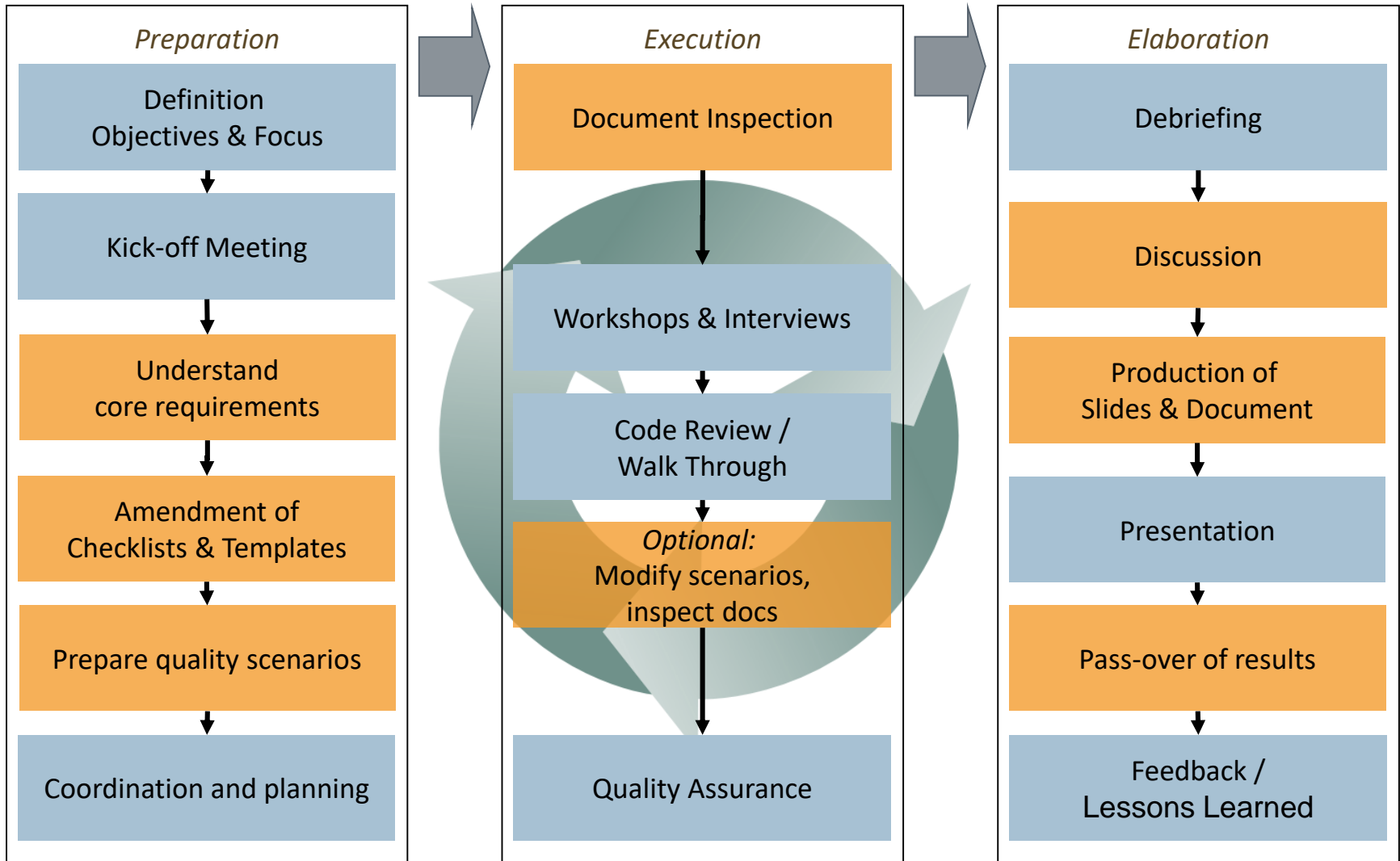
CodeCity, showing the packages as cities, classes as houses and methods as stories. The base area of a house is deferred from the number of attributes in a class.



Mixing the mode: multiple sources and multiple methods



The mixed-mode delivery process



Takeaways

- An architecture consists of concepts
 - These concepts have to be validated
 - This implies understanding the architecture
 - **Mixed-mode approach** is necessary
 - Architectural quality expresses itself on many tiers
 - Different sources and varying methods help form the big picture
 - Concrete issues and central questions steer focus
 - Paper doesn't blush
 - Talk to people!
- It is always preferable to assess a concept in creation rather than a productive system.



Thank you.

Juri Urbainczyk
Enterprise Architect

juri.urbainczyk@sn-invent.de

Twitter: jurbain
Tel. 0175 264 8042