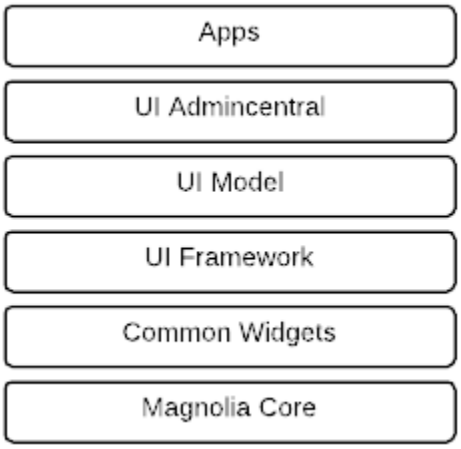
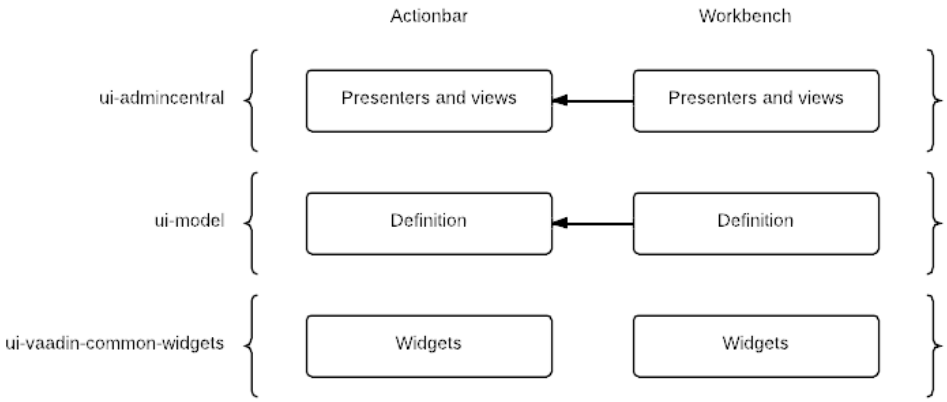
Concept - Alternative ui-project organisation

Abstract

In the current organisation we organise the code base by class roles keeping all definitions in the ui-model module, all vaadin components in ui-vaadin-common-widgets etc.

This creates a "horizontal split" where a part of the system, the actionbar for instance, is split over multiple modules. The classes that make up the actionbar is split in ui-model, ui-admincentral, ui-vaadin-common-widgets.





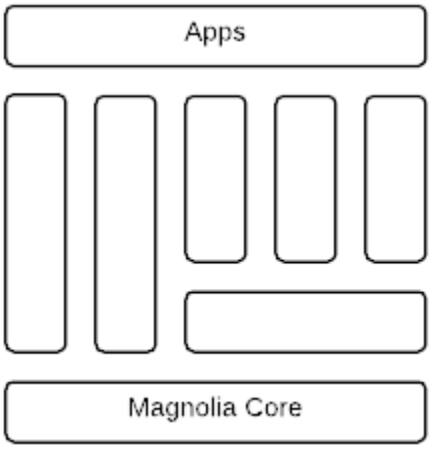
Problem

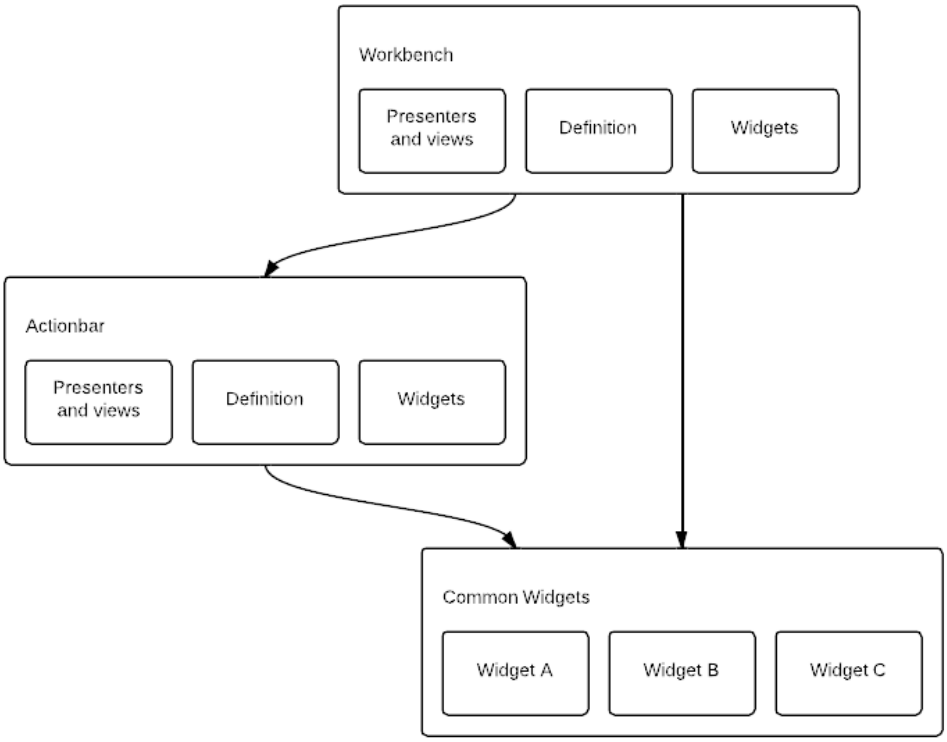
The "horizontal split" leads to a number of problems

- Circular dependencies, within for instance ui-admincentral its easy to introduce dependencies that lead to the code being entangled by features having mutual dependencies on each other.
- Code organisation, its hard to make out where to look for certain classes. When the code is split up in many locations its hard to keep track of it and hard to get a picture of how they fit together.
- Leads to spaghettification and creates big monolithic modules.

Proposal

The alternative is to make a "veritcal split" instead, where modules contain a certain part of the system, such as the actionbar or the workbench, and dependencies between these modules represents how the parts interact.





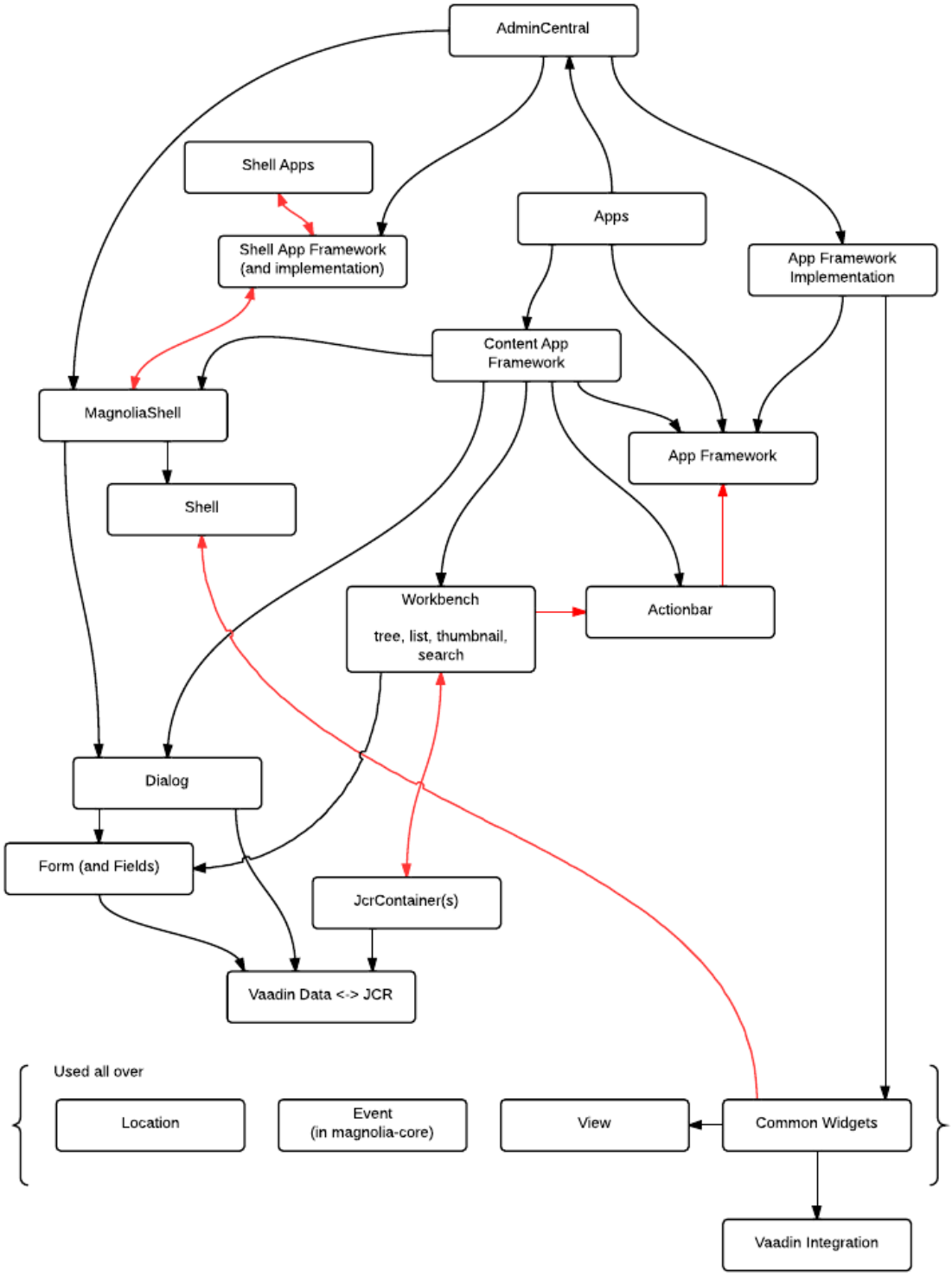
Benefits

The vertical split model has a number of benefits over horizontal split:

- All code relating to a certain function is in the same module
- It's impossible to introduce a circular dependency between modules
- It's easy to see how a module depends on another module and how they're assembled to produce the full system
- The division lends itself naturally to extension by introducing more modules.
- Modules stabilise as their functionality is completed

Dependency graph

The below graph gives a conceptual non-complete view of how functionality in ui-project interact. Red lines indicate circular dependencies or otherwise problematic dependencies.


AdminCentral
Shell Apps
Shell App Framework
(and implementation)
Apps
App Framework
Implementation
Content App
Framework
MagnoliaShell
App Framework
Shell
Workbench
tree, list, thumbnail,
search
Actionbar
Dialog
Form (and Fields)
JcrContainer(s)
Vaadin Data <-> JCR
Used all over
Location
Event
(in magnolia-core)
View
Common Widgets
Vaadin Integration