

Modular Architecture

Liferay's Modular Architecture Explained

Raymond Augé <raymond.auge@liferay.com>

 @rotty3000 | #lrnas2014 #modulararchitecture

Outline

⌕ As it Was

⌕ As it Is

⌕ As it Shall Be

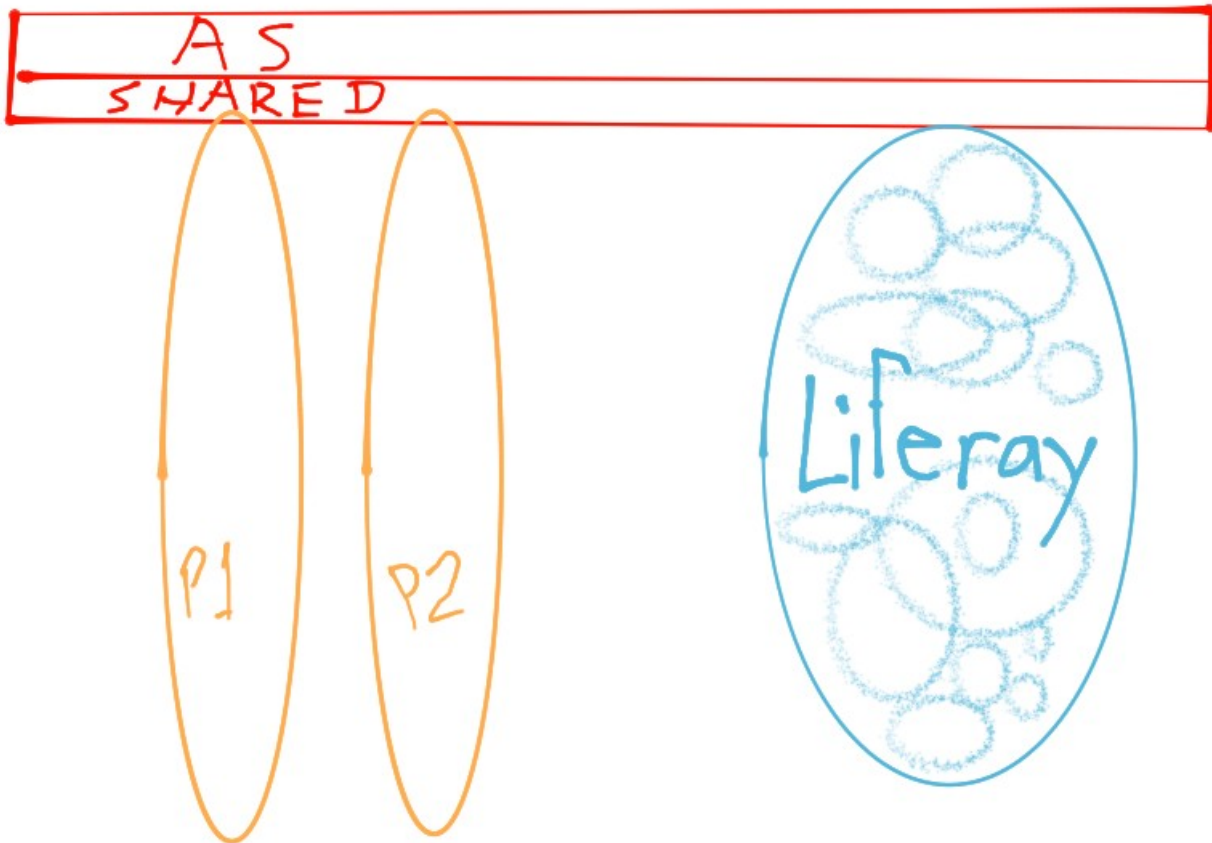
⌕ Closing

⬅ As it Was - Thousands of features



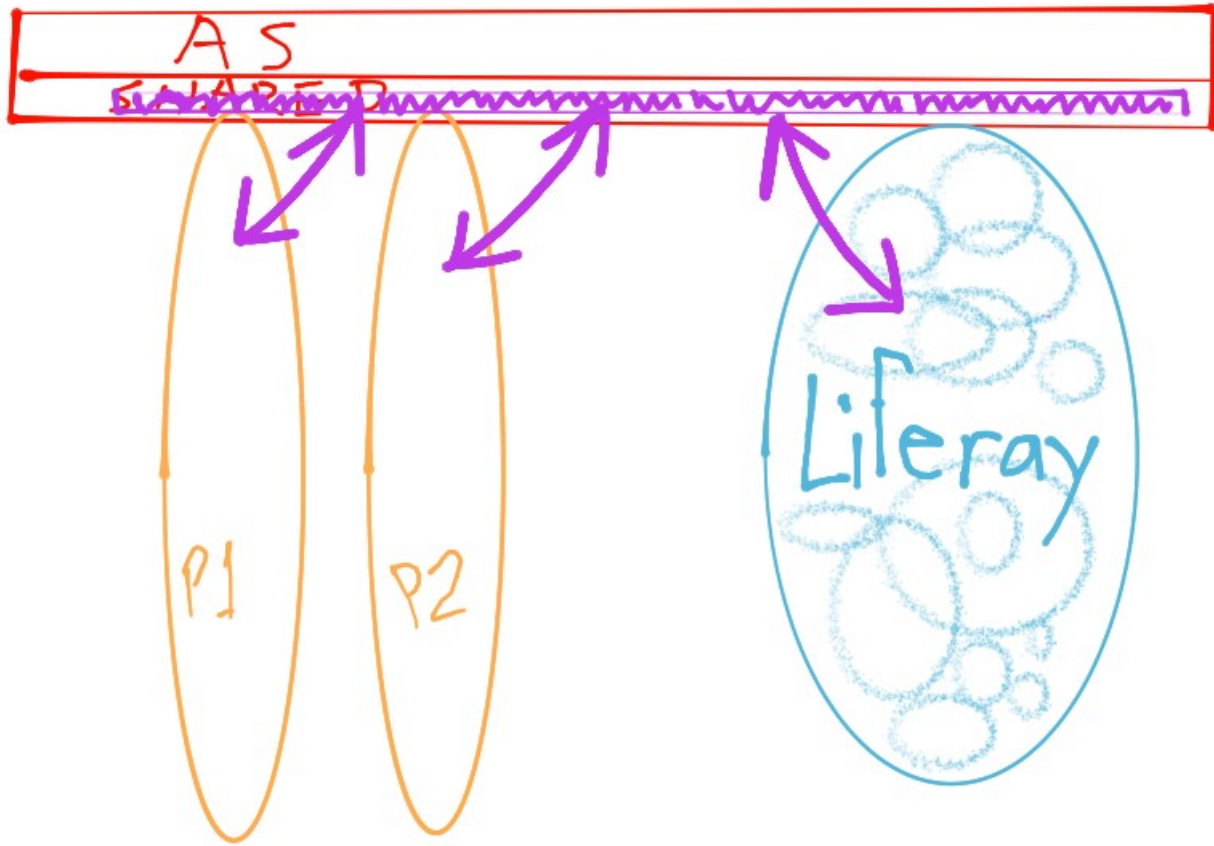
- Tightly coupled
- No clear boundaries
- All of nothing

⬅ As it Was - The ties that bind us...



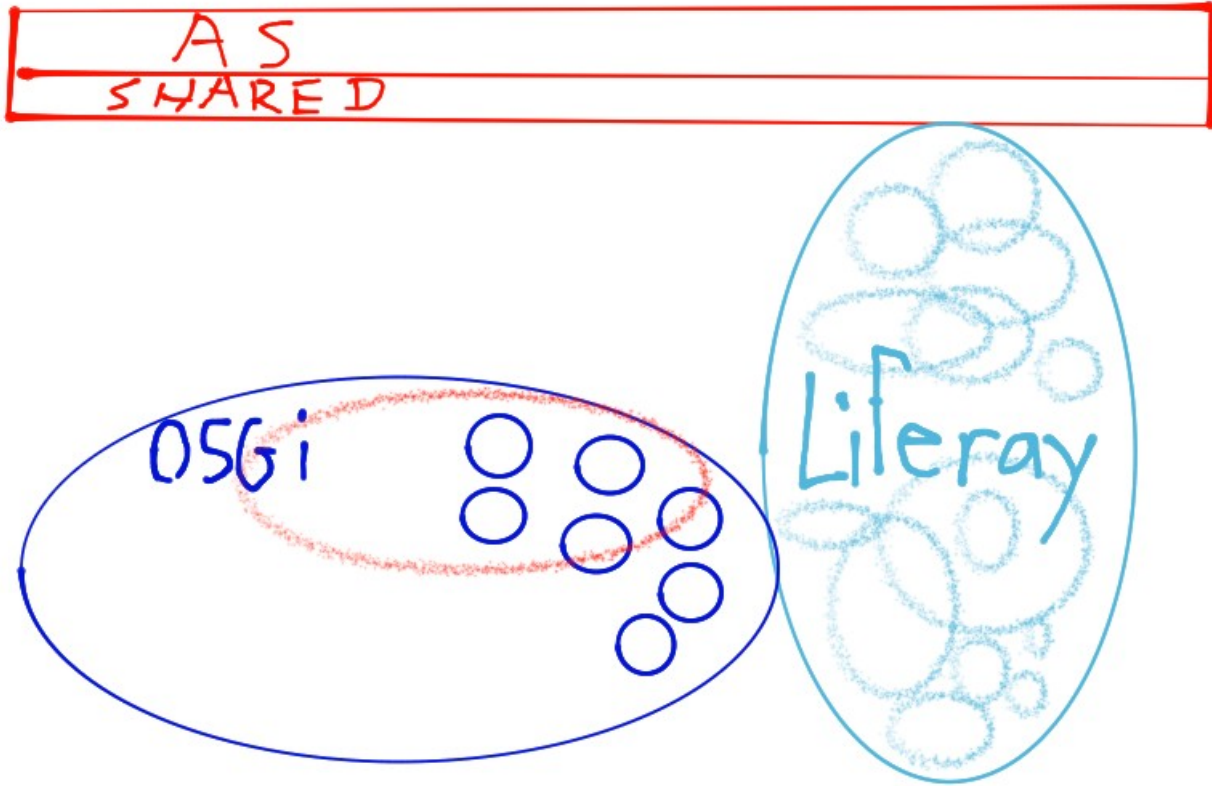
- At the mercy of the app server
- Plugins are peers
- Start order issues
- Dependency issues
- Security issues

⬅ As it Was - Bridge Over Troubled Waters



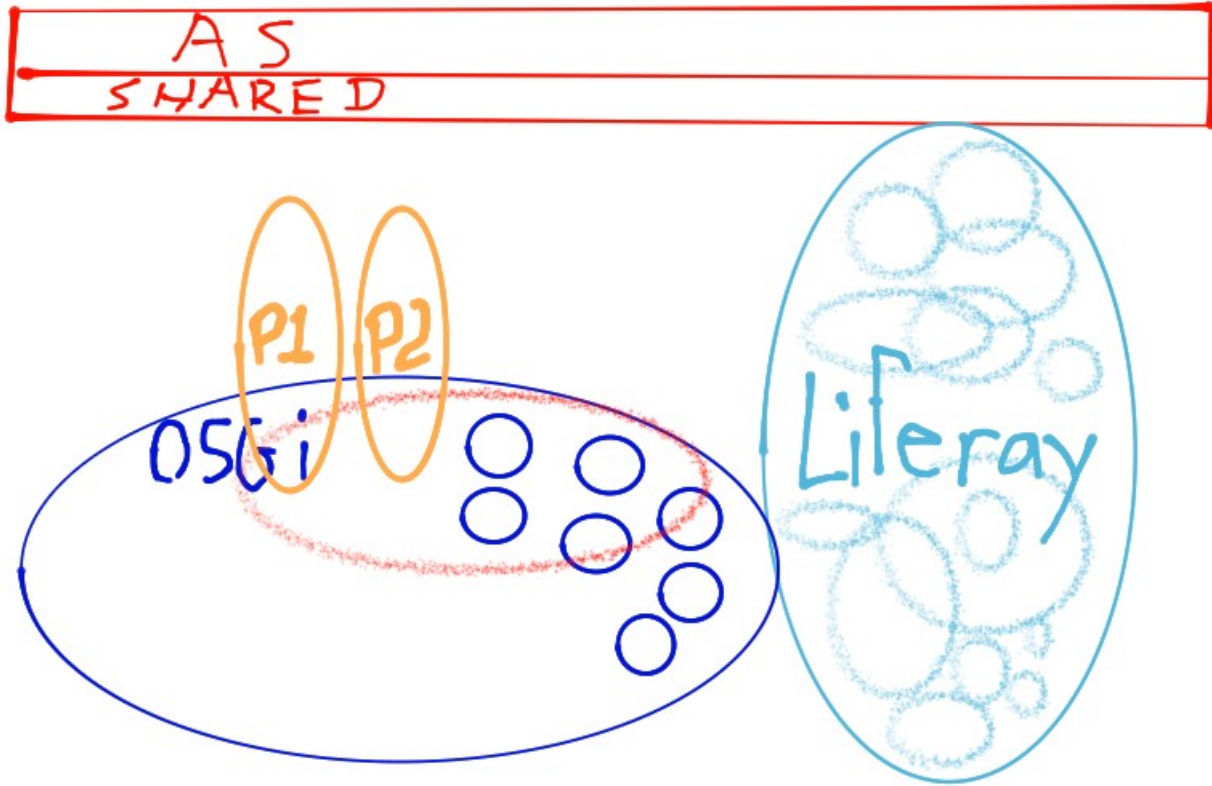
- Push as much as we can into *shared*
- App server idiosyncrasies
- Protect impls through indirection
- Hundreds of wrapper APIs
- Statics became the norm

⬇ As it Is - Take Back Control



- Create a plugin runtime under Liferay's control
- Implement infrastructure support
- Emulate some of the App Server features

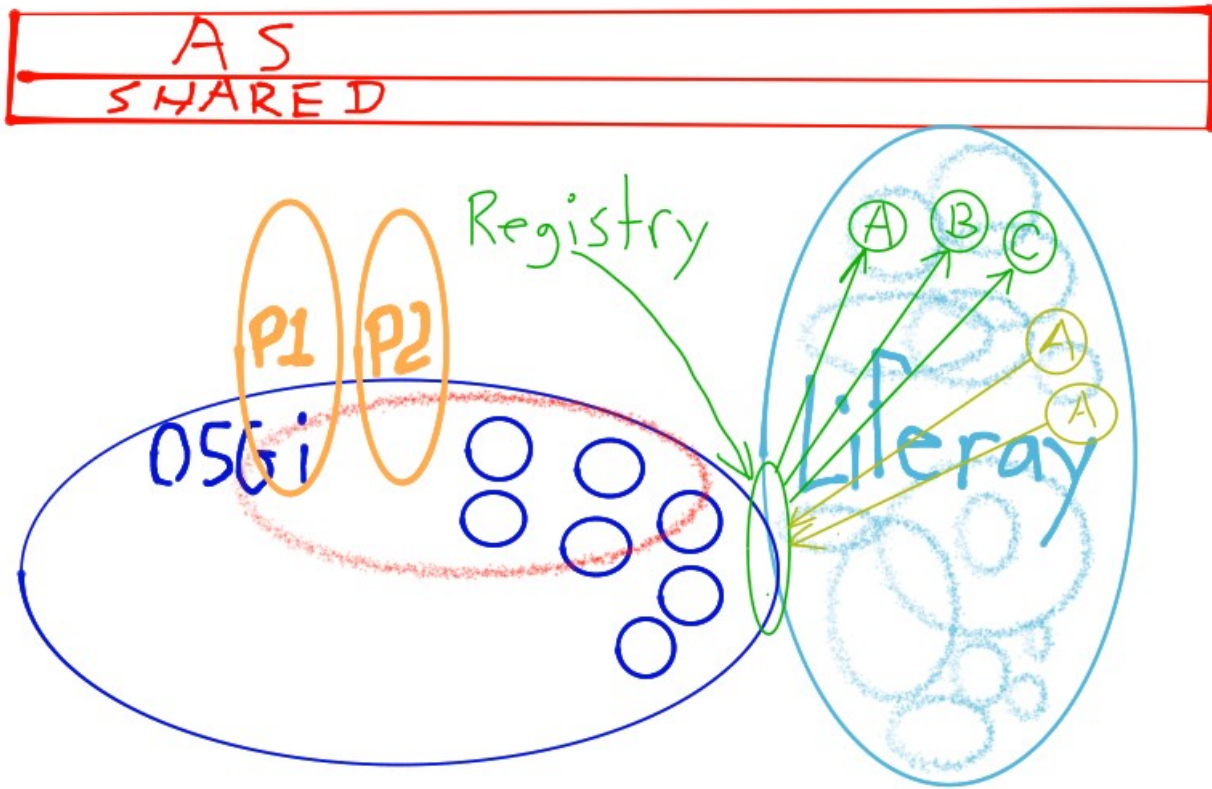
⬇ As it Is - Don't turn your back



- Make sure that legacy apps (WARs) *can* still work > provide for
 - Auto conversion to OSGi (WAB)
 - JSP support in OSGi
- In the process > Make a few enhancements
 - Implement better dependency management

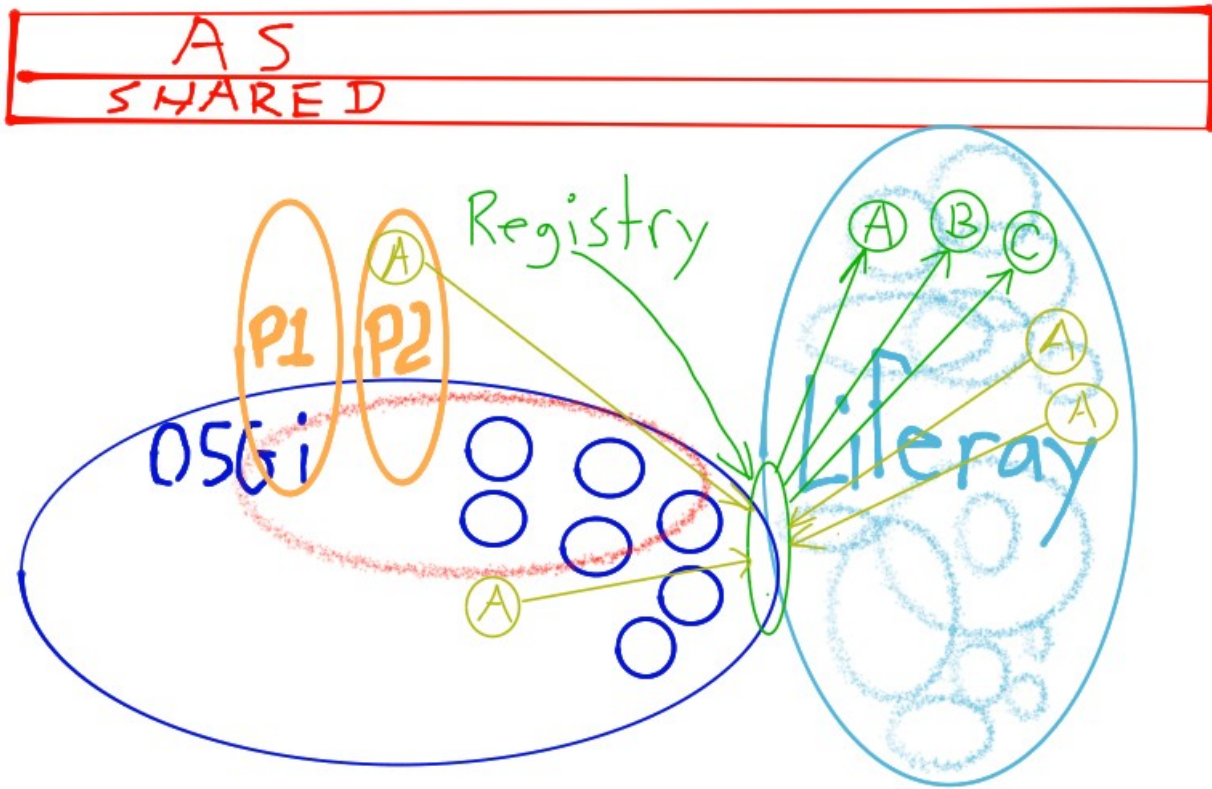
- Publish SB services as OSGi services
- Eliminate SB service jar copying (go direct, don't need CLP)

⬇ As it Is - Tear Down the Walls



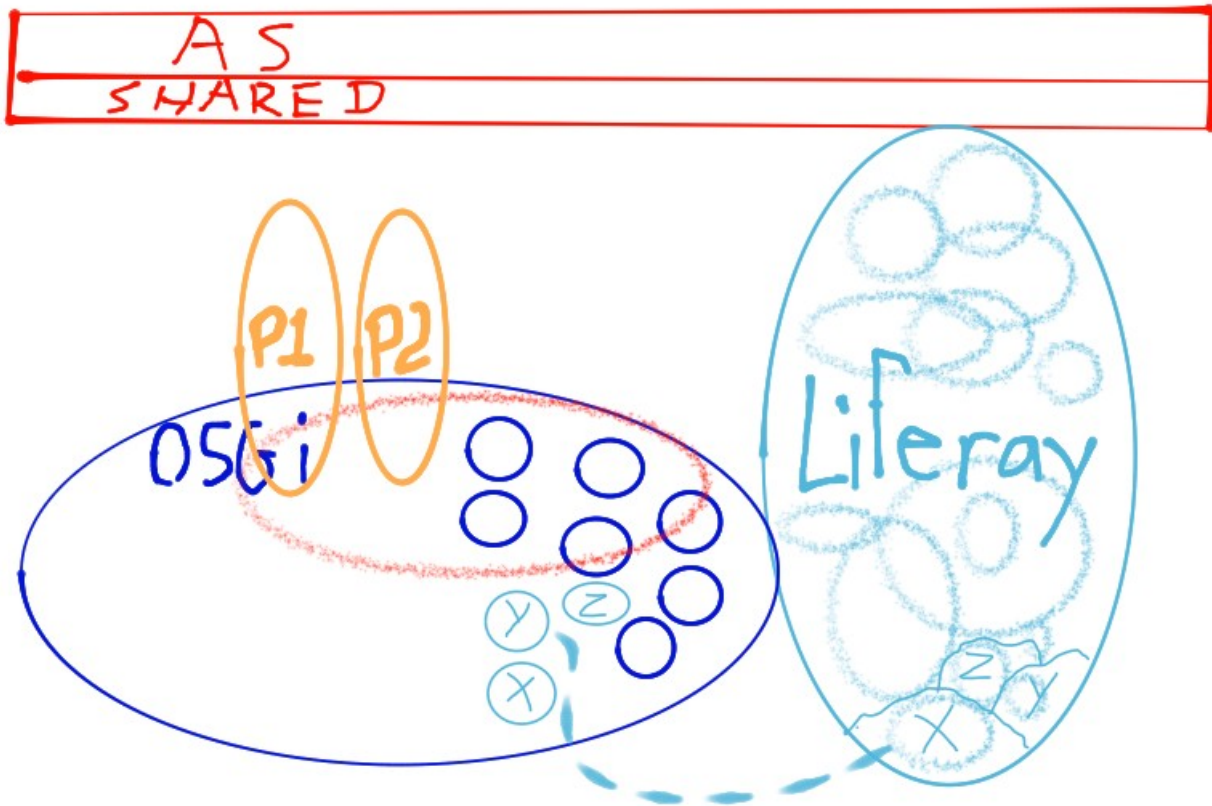
- OSGi decoupling is primarily based on the service registry
- Simplify integration point logic through the Registry

⬇ As it Is - Tear Down the Walls



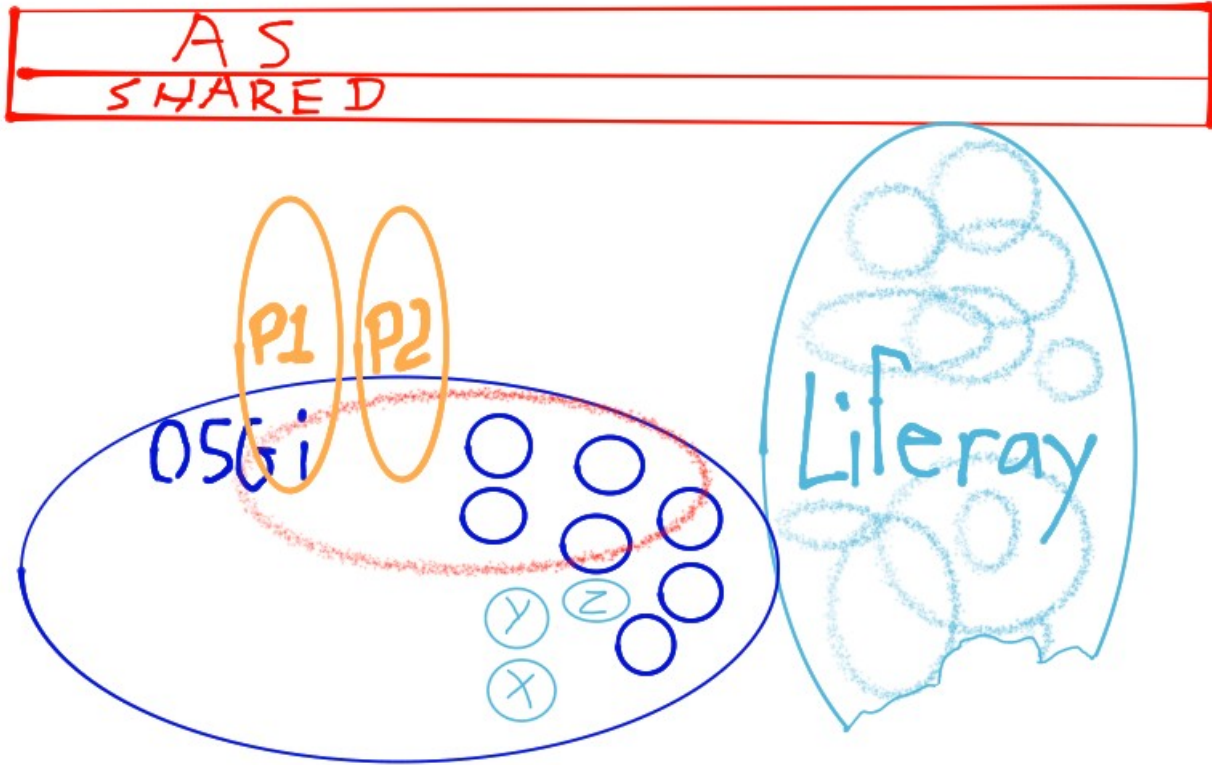
- Registry opens the door for providers from any source
- Registry allows integration points to be placed anywhere

⬇ As it Is - Breaking it Down



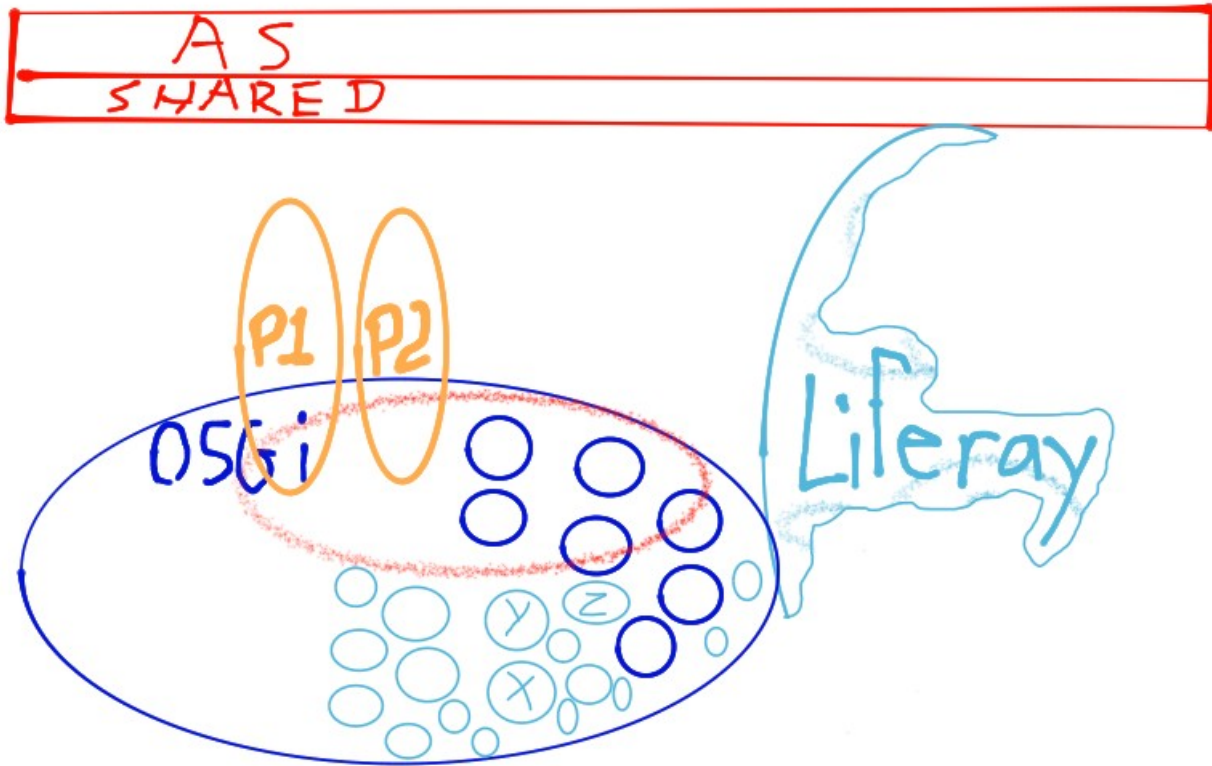
- Refactoring core pieces into modules

⬇ As it Is - Breaking it Down



- Shrinking the monolith
- Bringing optionality

➡ As it Shall Be - Strong Foundation



- Core of essentials
- Increase re-use
- Smaller physical/memory footprint
- Easier configuration

↑ Closing



- Monoliths are hard to maintain and have higher production costs.
- Modularity promotes isolation which promotes re-use which promotes innovation.
- Modularity reduces coupling which reduces maintenance costs which leads to satisfied users.

Liferay is doing whatever it can to increase productivity, reduce production and maintenance costs and to generally get out of your way while getting you to your goal faster.

 **Thank You!**

- **Questions?**