



Future proof your applications with API Regions

Karl Pauls | Adobe

David Bosschaert | Adobe



About the Speakers

David is a Senior Software Engineer at Adobe. He has been working on OSGi specifications since 2007 and is an Apache committer and member.

Karl works as a Computer Scientist at the AEM Cloud Foundation team at Adobe. He is a member of the ASF working on Apache Sling and Apache Felix.

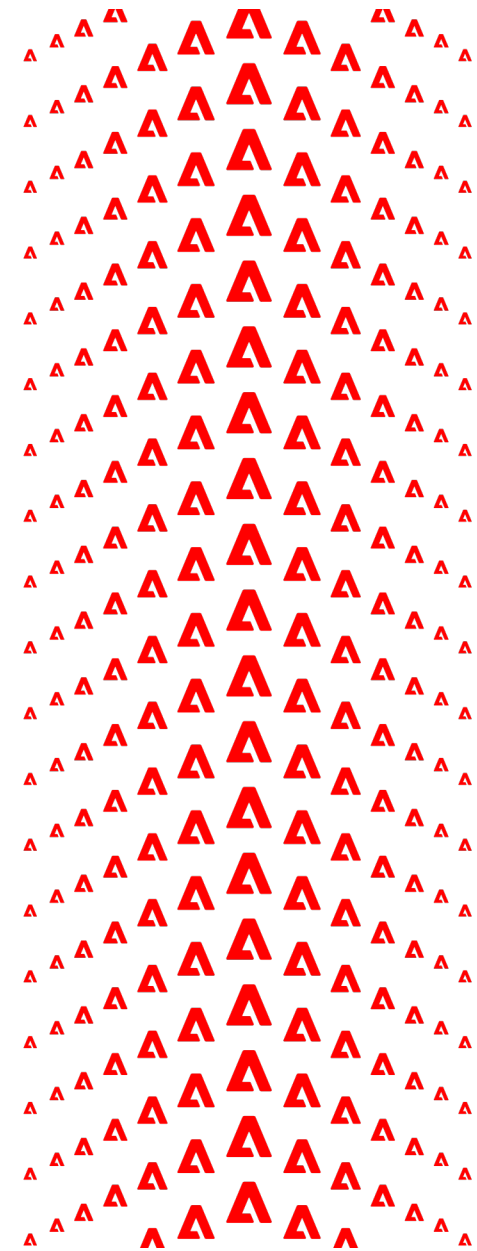
Agenda

Building and maintaining a platform

Introducing API Regions

Designing the Regions

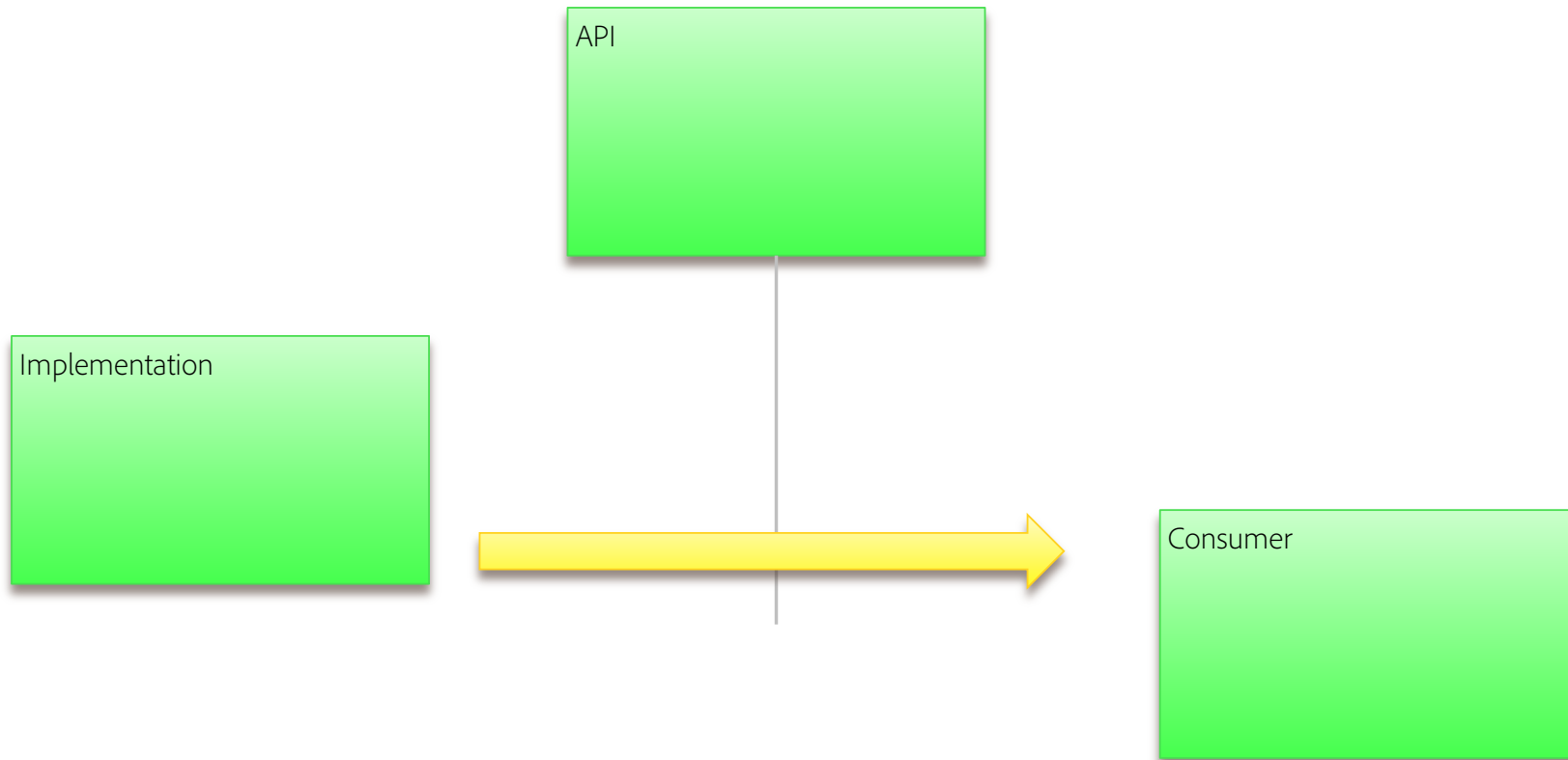
Build and Runtime Validation

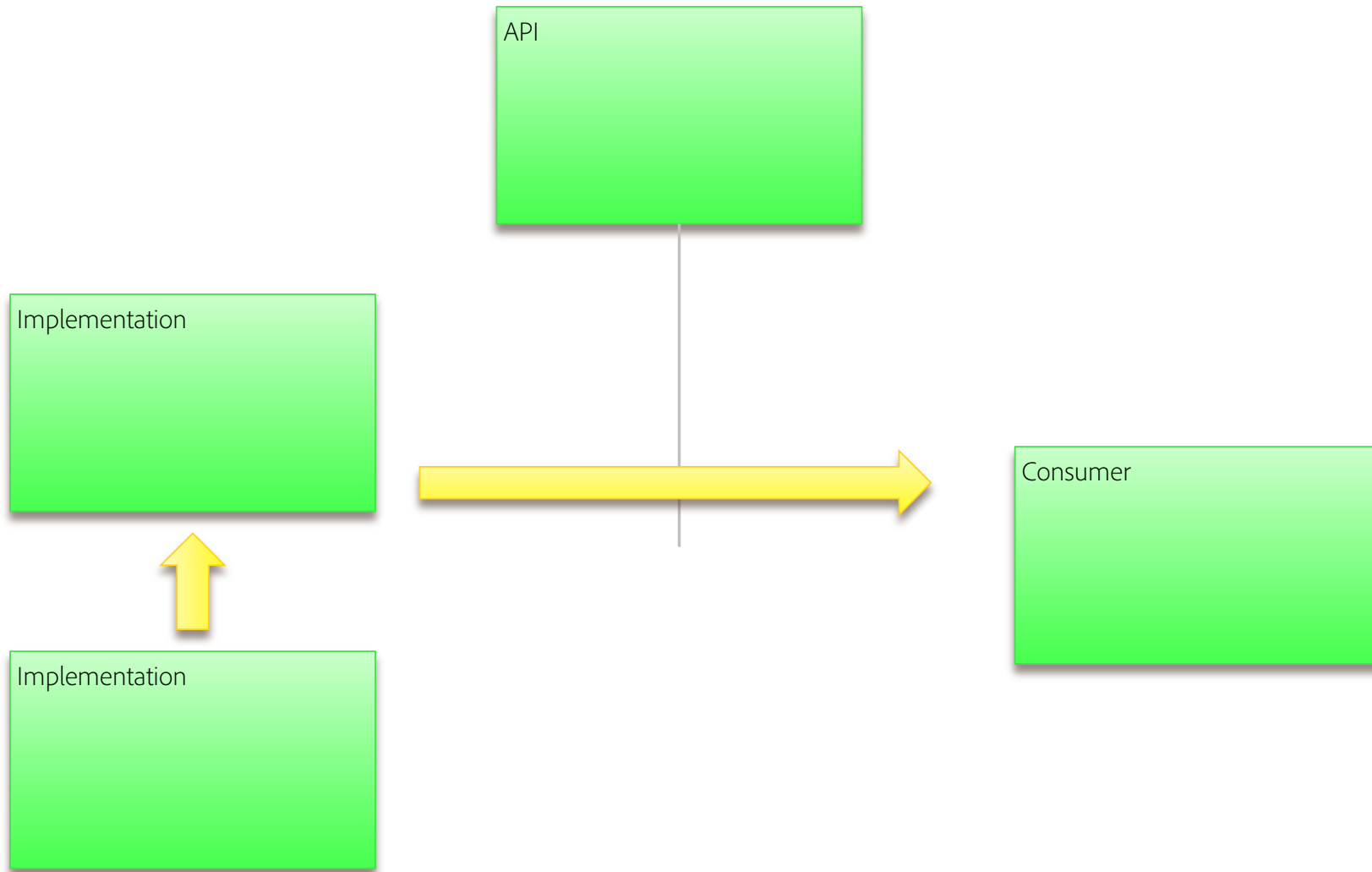


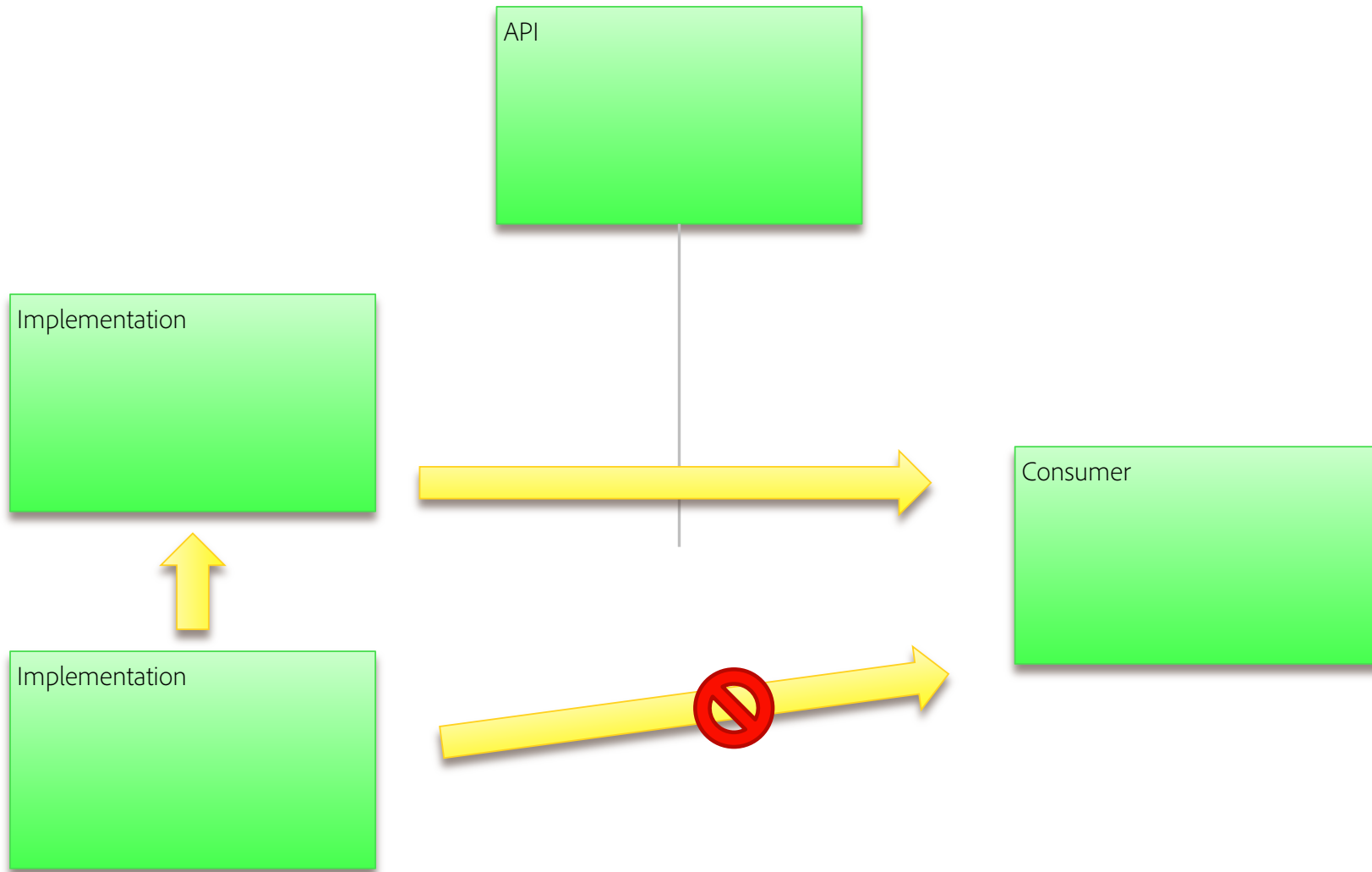
Building a platform

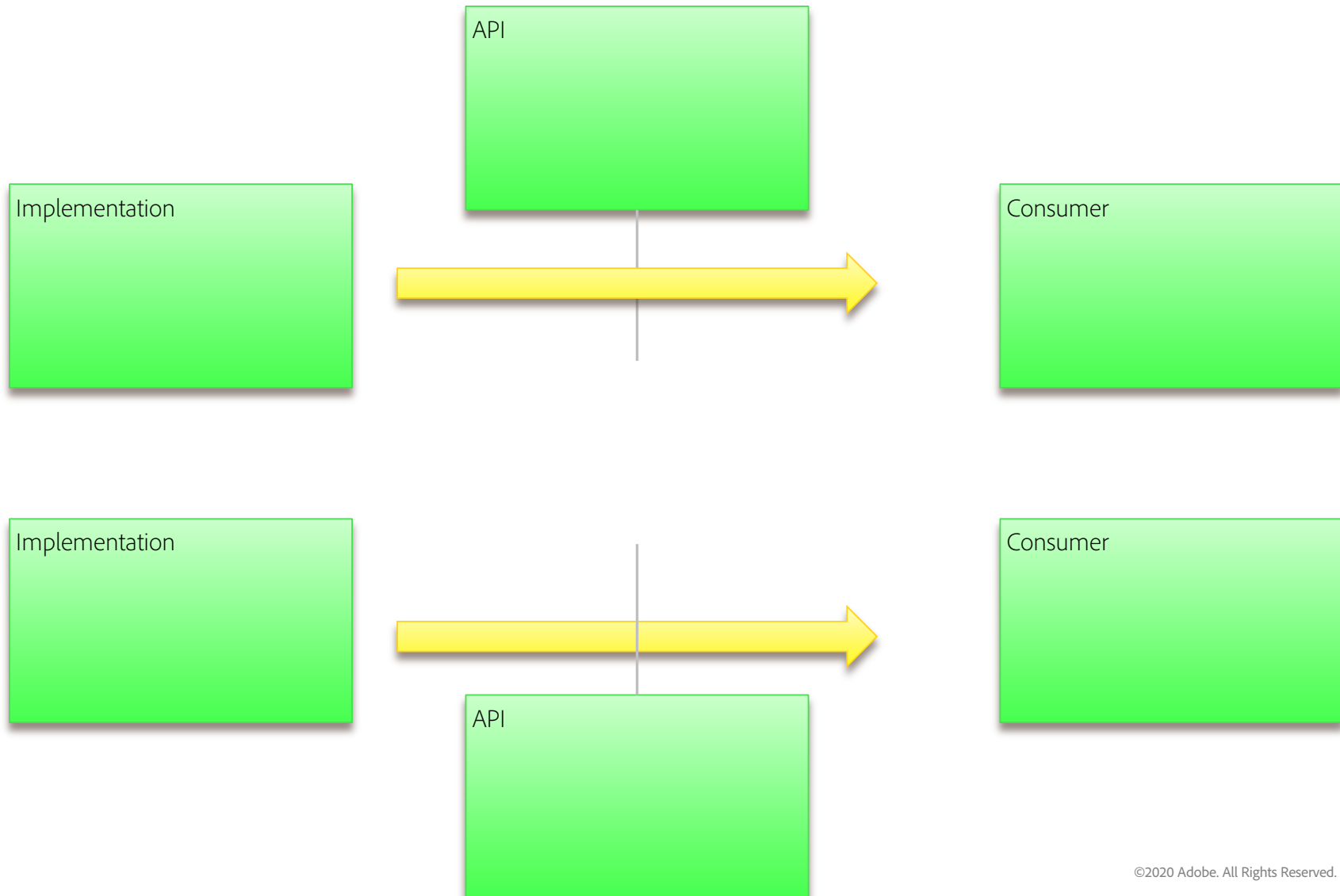
- A platform typically needs to expose an API
- Likewise, it needs internal API to function that shouldn't be available to user
 - Desirable to be hidden completely
- Over time, deprecation and backwards incompatible changes require some grey areas
 - Mistakes do happen
- Optionally available API for certain cases
 - Extensions/Add-ons











Giving users Freedom

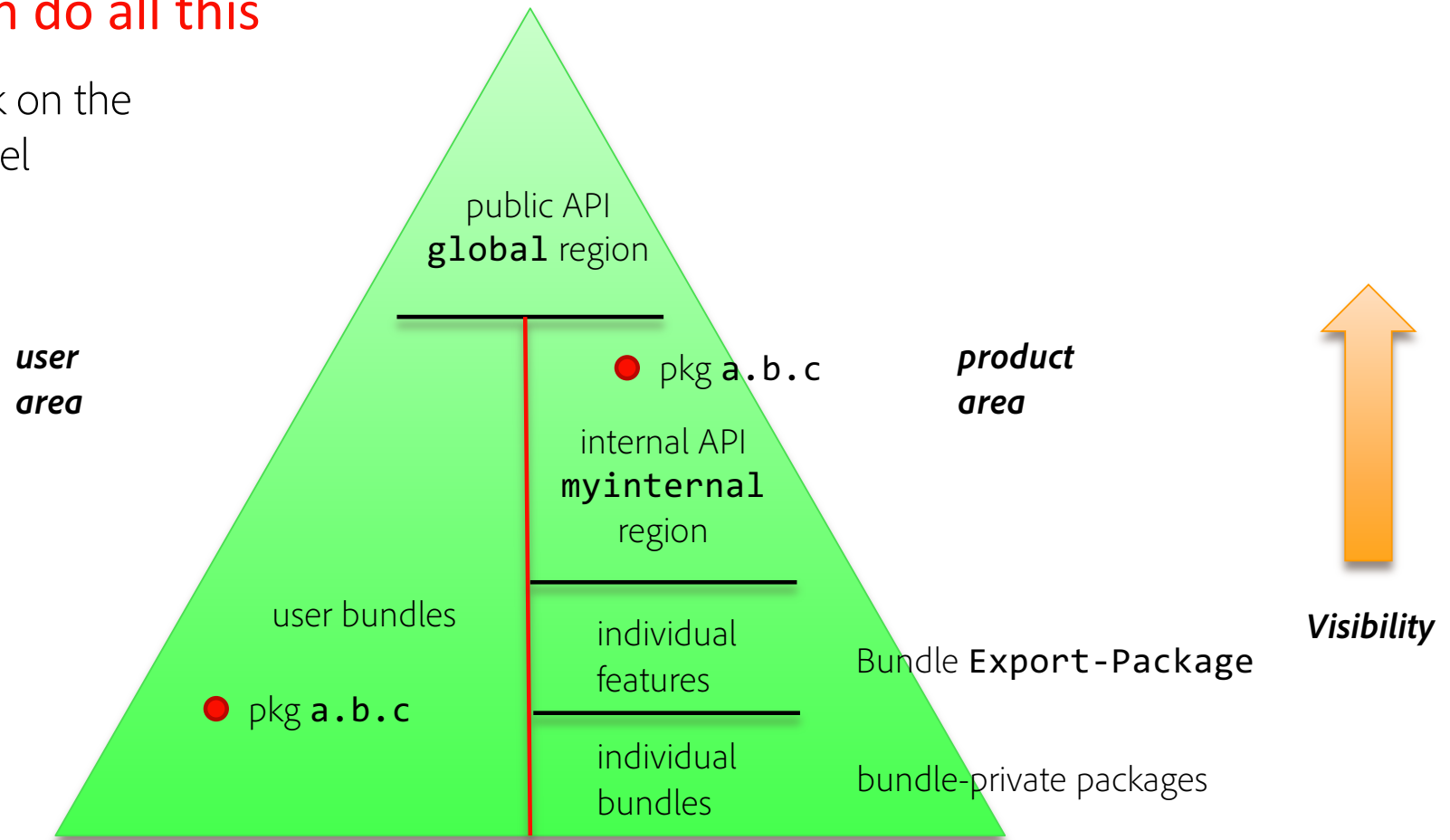
- OSGi
 - Explicit sharing of packages only
 - Versioning of shared packages
 - Packages can be constrained further
 - Multiple versions at the same time

Keeping the platform Maintainable

- Exposing various levels of API is possible using OSGi
 - Conventions and documentation needs to be used
 - Avoiding problematic cases is hard
 - Backwards compatibility can become a major concern
- Better solution:
 - Enforce the regions at runtime and provide checks at build time
 - OSGi subsystems
 - API Regions

API Regions can do all this

- API Regions work on the Java Package Level



API Regions and Feature Models

Feature Model

... extensions ...

```
"api-regions": [{  
  "name": "global",  
  "exports": ["org.myapi"]  
}, {  
  "name": "myinternal",  
  "exports": ["org.myutil"]  
}]
```

Feature Model

... extensions ...

```
"api-regions": [{  
  "name": "global",  
  "exports": ["my.otherapi"]  
}]
```

Feature Model

... extensions ...

```
"api-regions": [{  
  "name": "myinternal"  
}]
```

- Everyone can see APIs in the **global** region
- Opt-in to see a specific region

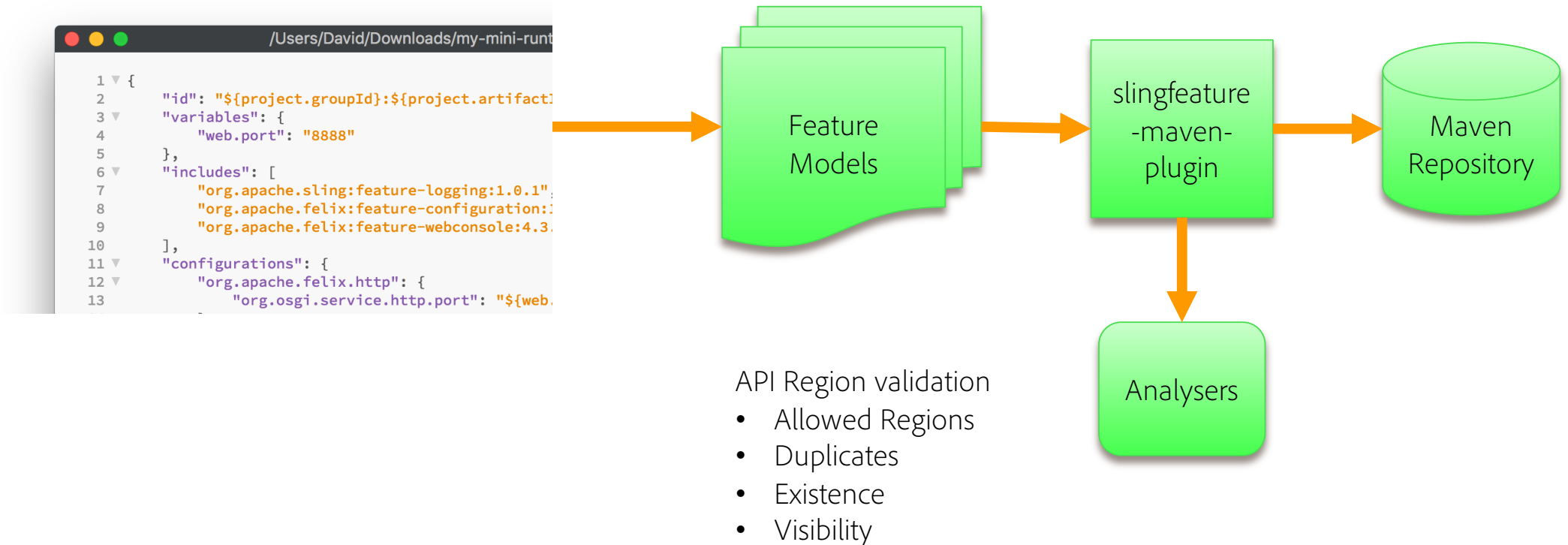
API Regions – Usability

- API Regions are the concern of the Platform Designer
- Users generally don't need to be aware
- Stuff just works

Designing the Regions for your Platform

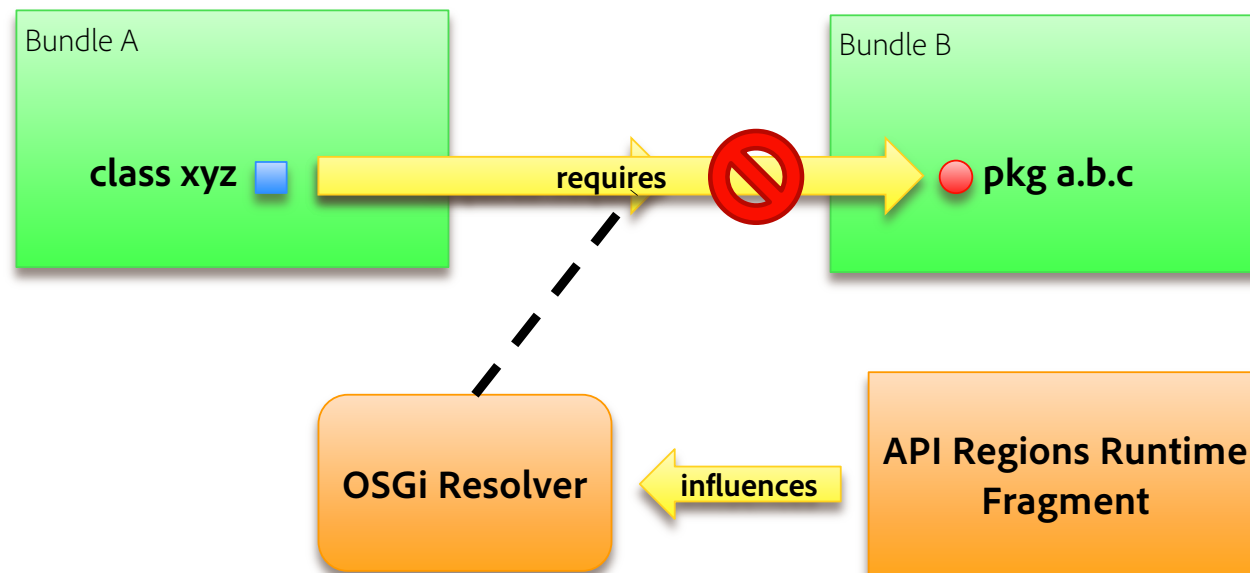
- Everyone can see the **global** region
- Keep your **global** region as small as possible
- Less API -> easier to understand
- Less API -> less to maintain
 - more opportunity for refactoring
- Put as much as you want in the **myinternal** region
 - make it lean or rich
- It's just for your own platform
 - if it changes it doesn't affect users

Build time Validation



Runtime Enforcement

OSGi Framework



Runtime enforcement is handled during package resolution

API Regions Runtime – Debugging

Adobe Experience Manager Web Console

Services

Main OSGi Sling Status Web Console Log out

Services information: 1 service(s) in total.

Filter: (objectClass=*ResolverHookFactory) Apply Filter

Id	Type(s)	Bundle	Ranking
7	[org.osgi.framework.hooks.resolver.ResolverHookFactory] bundles.properties features.properties idbsnver.properties regions.properties service.bundleid service.scope sling.feature.apiregions.default	org.apache.felix.framework (0)	0

Services information: 1 service(s) in total.

DEMO

