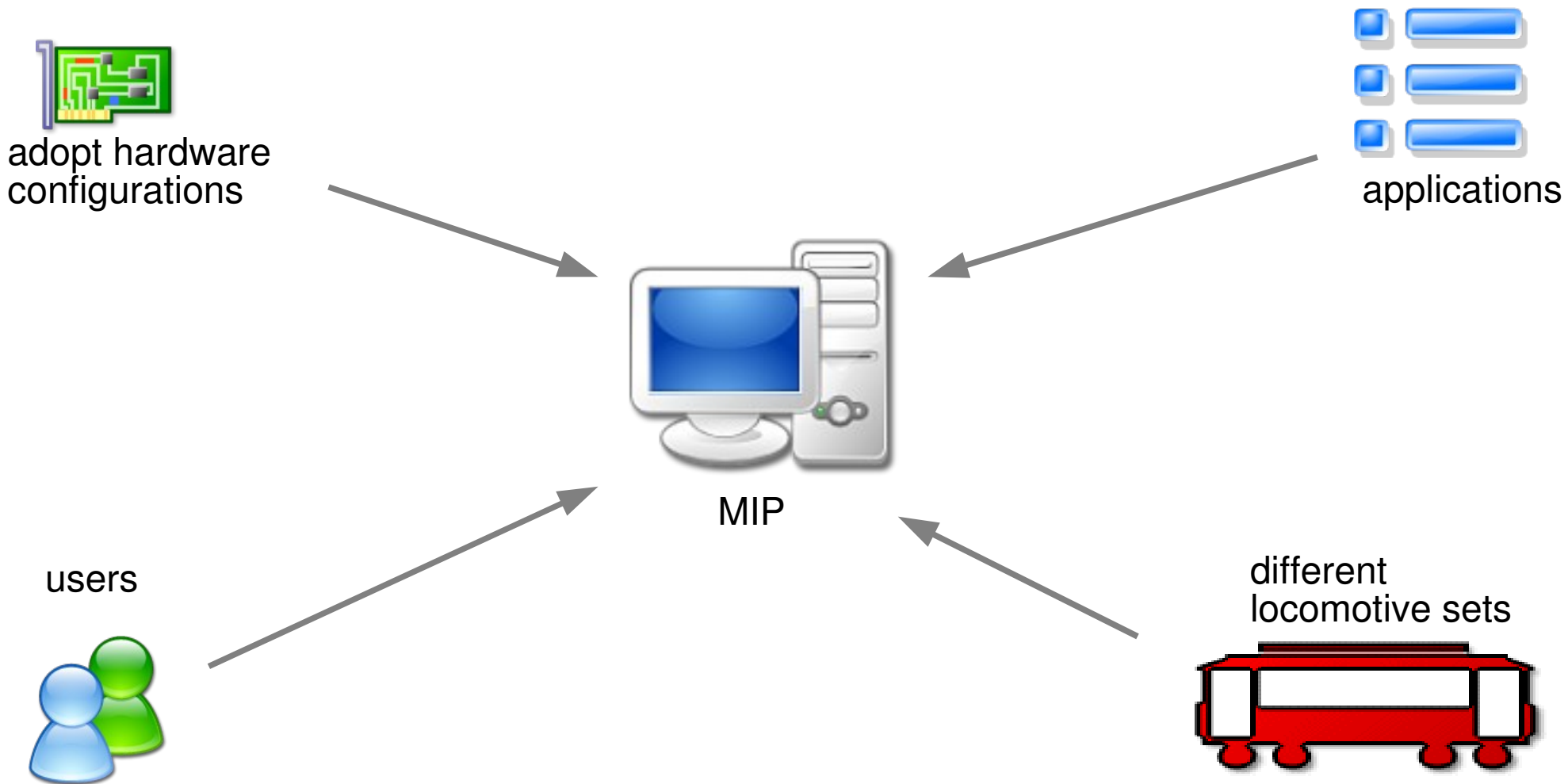
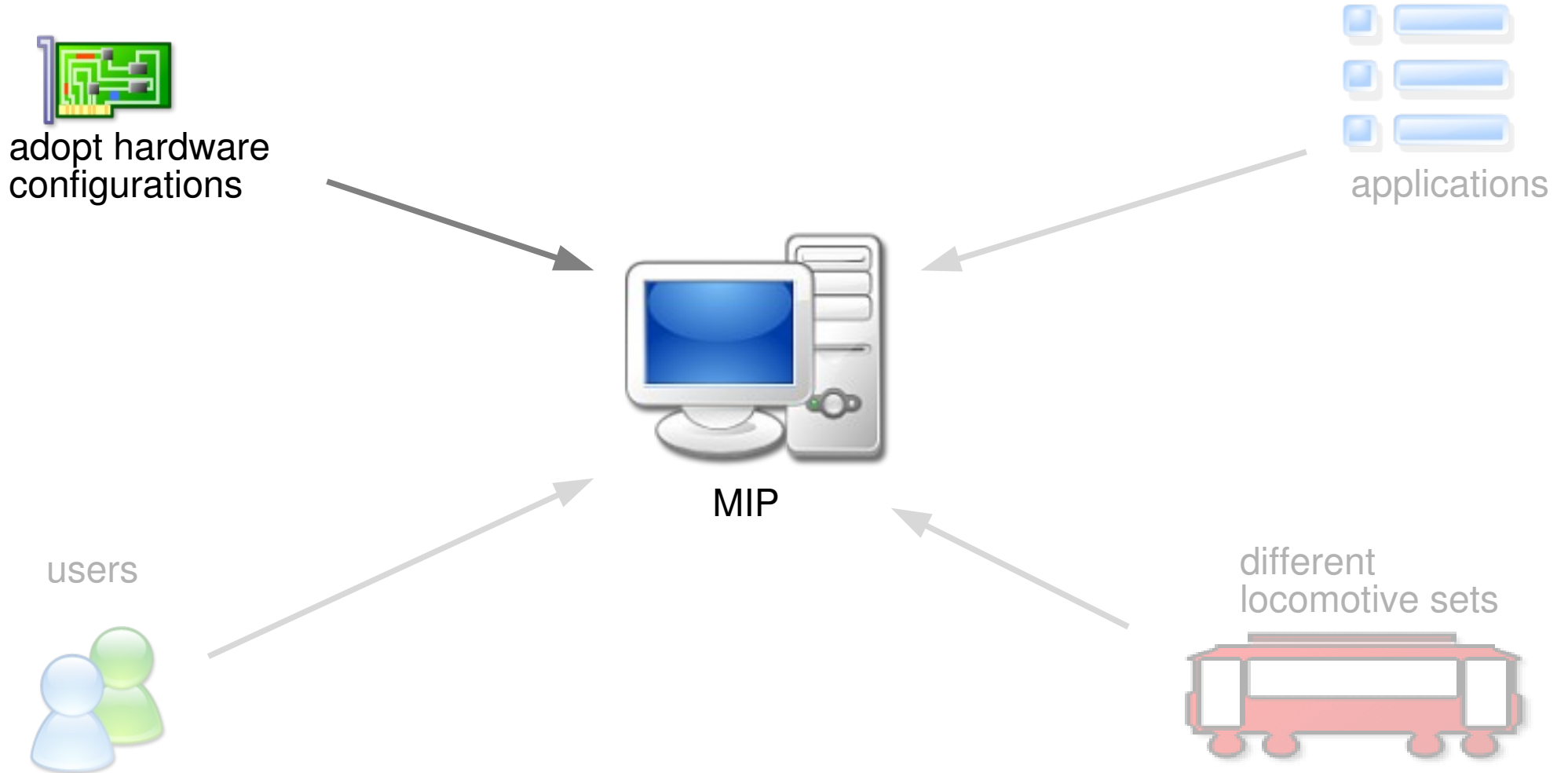


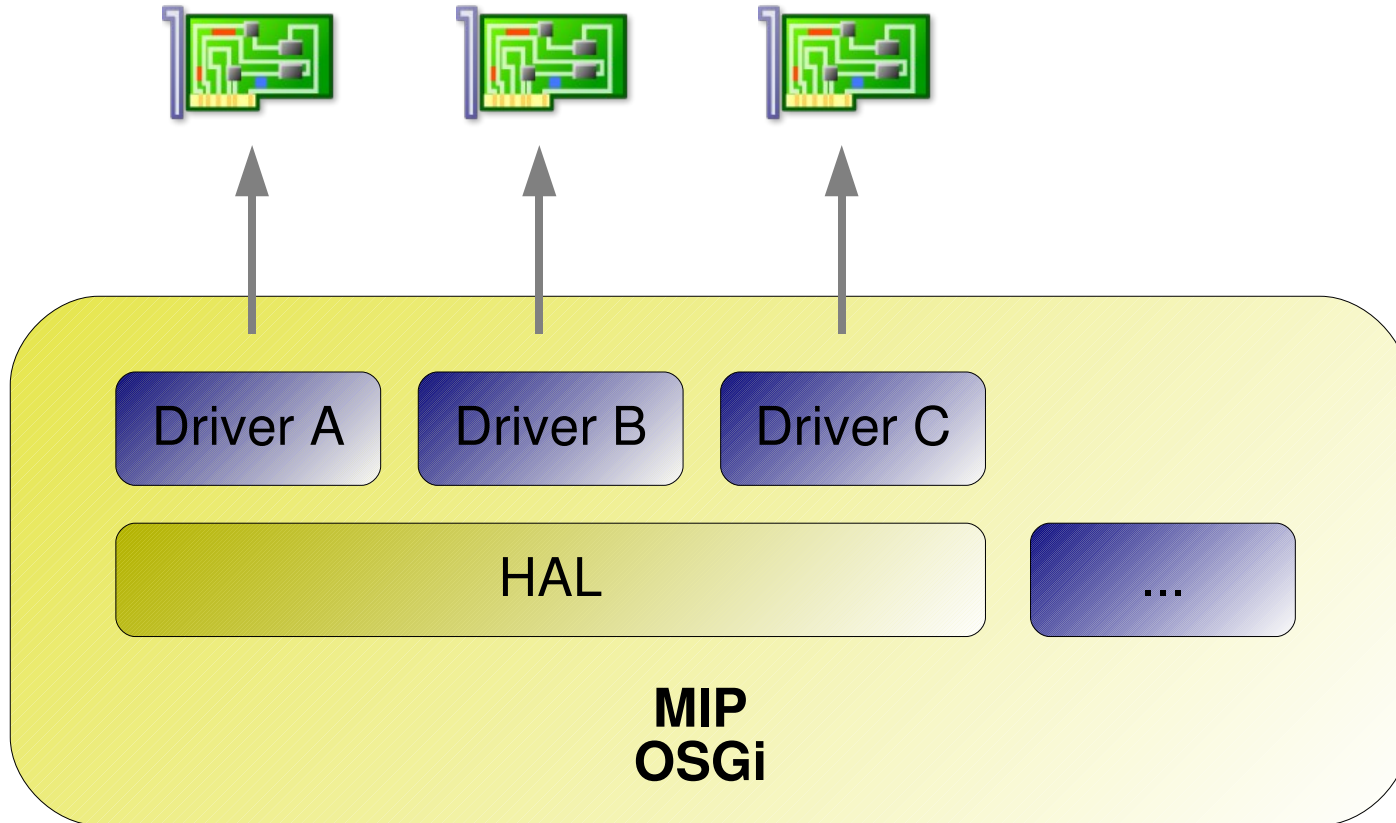
Technical Details

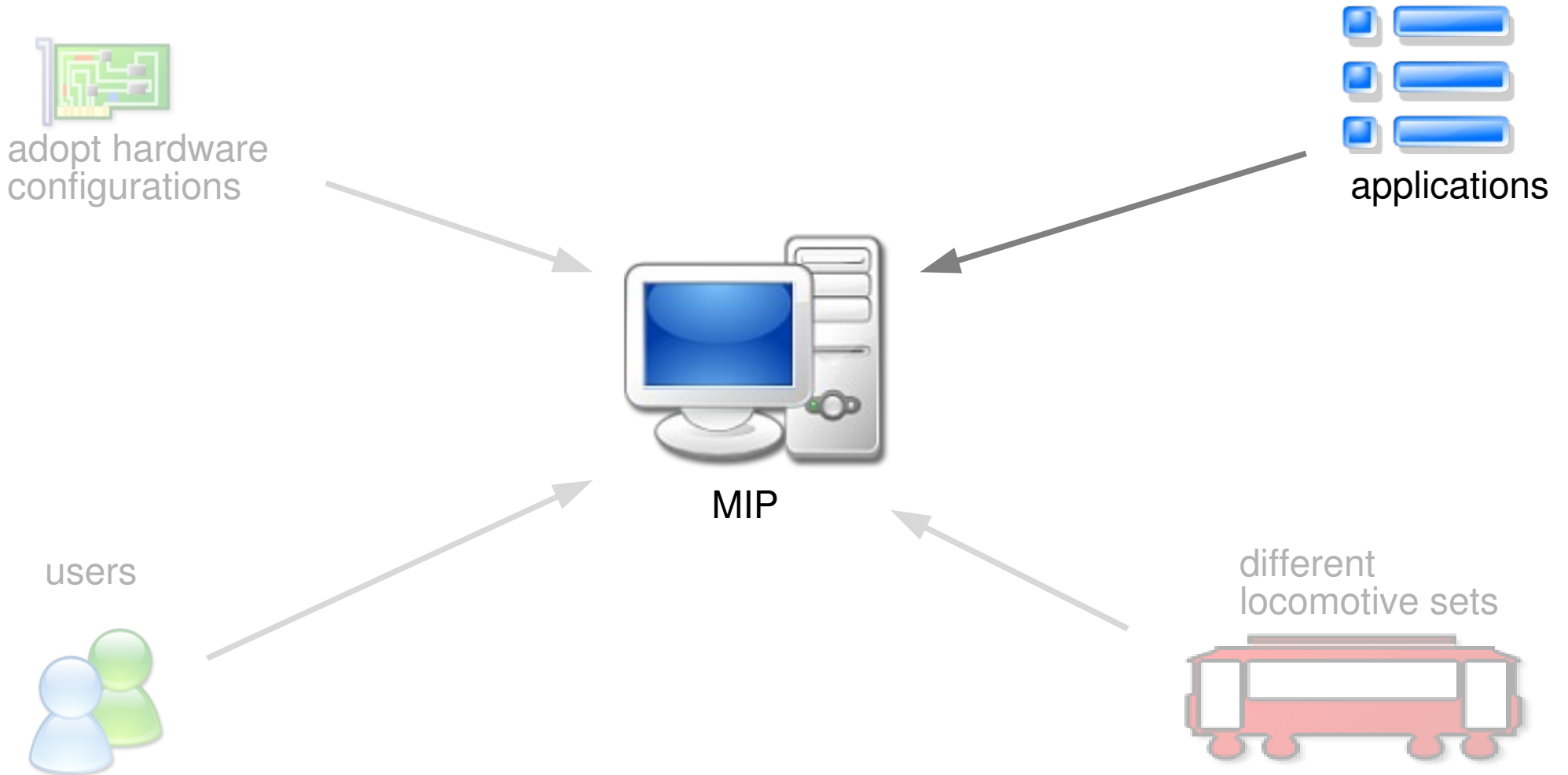




- The MIP has to interact with different kind of devices
 - Displays
 - Loudspeaker
 - ...
- It must be possible to use N devices of the same type, but from different vendors
 - Display from vendor 1 (HTML)
 - Display from vendor 2 (plain text)

- Hardware abstraction layer to decouple the applications from the hardware
- Devices provide drivers to the HAL
- Driver = Bundle
- Clients access the HAL instead of the hardware driver directly
- Simple, but already an effective mechanism for abstraction
- Replace the bundles depending on the actual hardware
- No need to change the clients





- Applications are based on existing functionality
 - Connected devices, Basic services (e.g. GPS), ...
- Applications can add new functionality to the system
 - Make the consumption as easy as possible
- Application can reuse existing applications
 - Composition

- Applications provide functionality in the form of services
 - The OSGi / POJO kind of service
- Applications consume services
 - LDAP filters are great!
- Services everywhere
 - Application services
 - Infrastructure services
 - OSGi-specific services

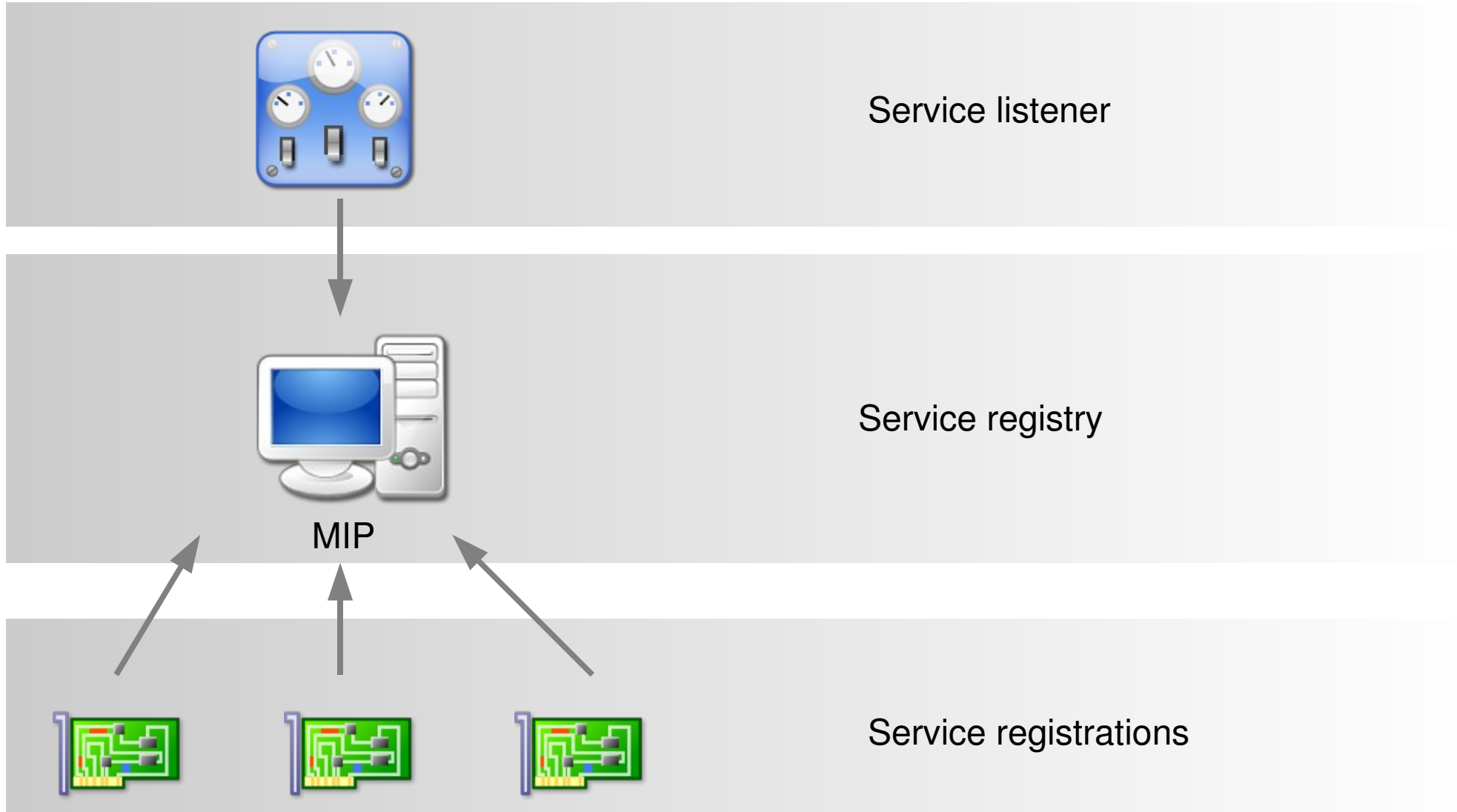
- Specify services for well-known domains
- Register service in the service registry
- Easy to extend
- Decouples provider and consumer

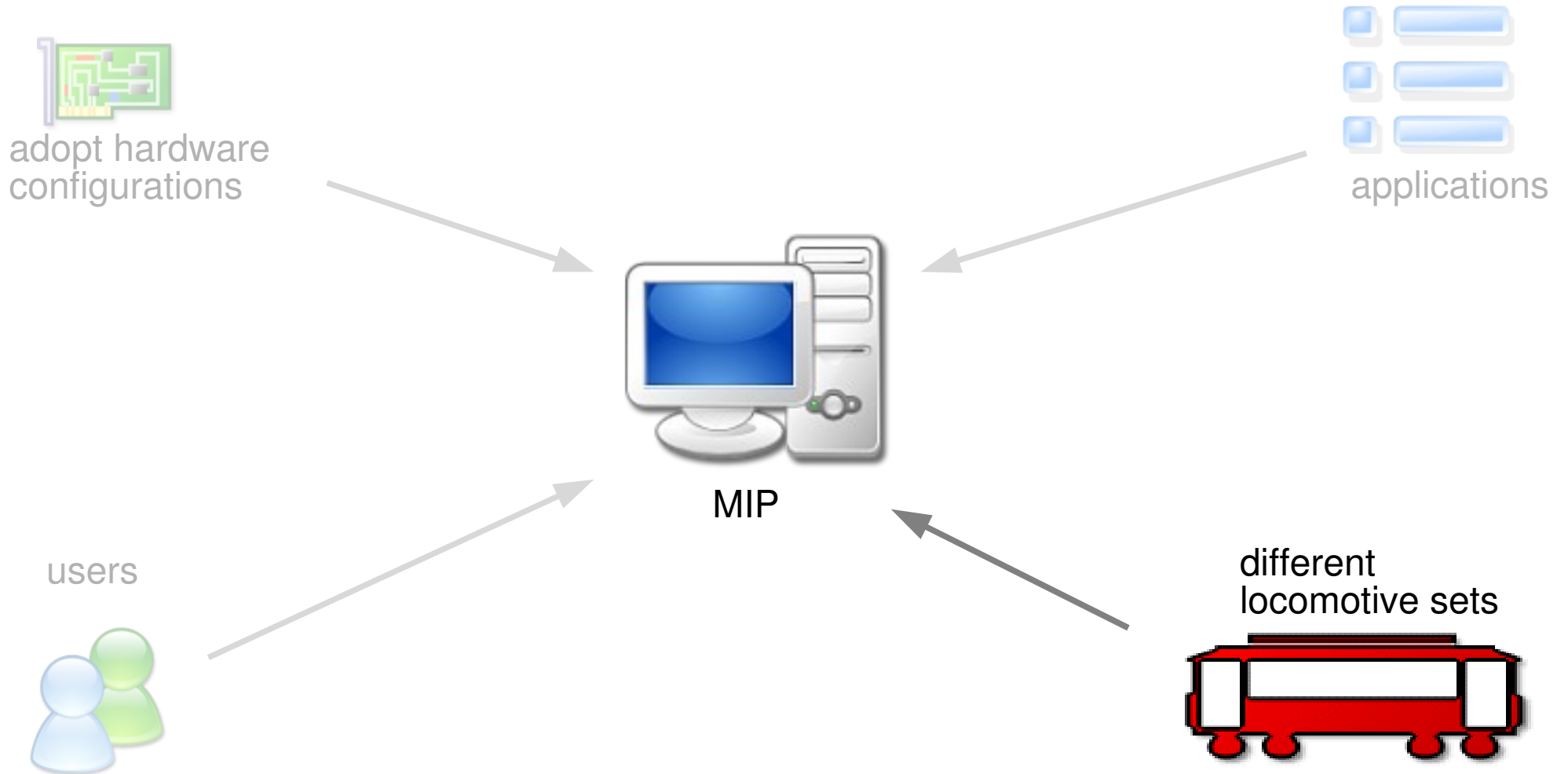
TrackService

```
+ getDistance()  
+ getSpeed()
```

- Scenario: Obtain status information from all connected devices
- Drivers register services to provide the necessary information
 - All use the same interface type!
- Surveillance module queries the service registry to obtain the status informations from the devices
- Whiteboard Pattern

```
BundleStatus  
+ getStatus()  
+ getVersion()
```

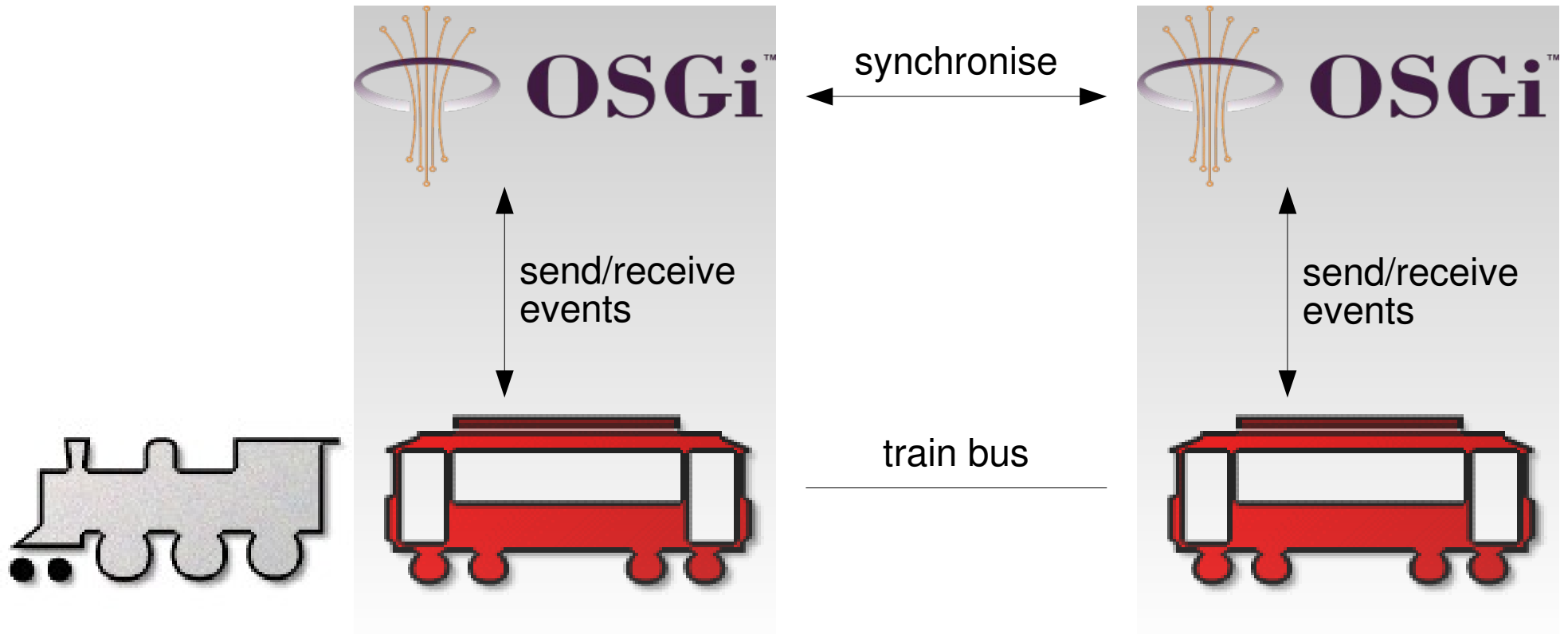


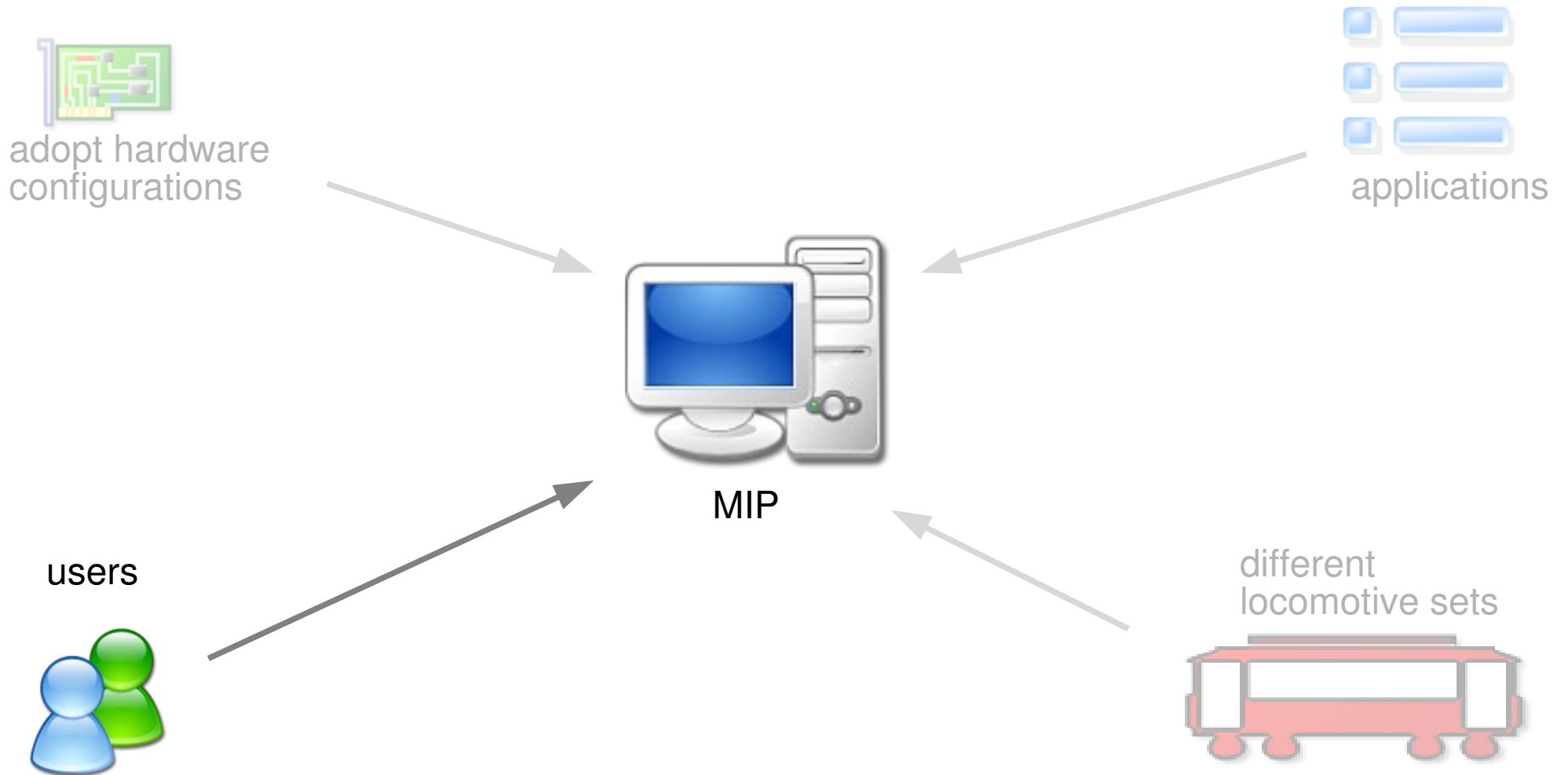


- The MIP will be used in several different locomotive sets
- The position and number of the wagons can change all the time
 - 1 engine vs. 2 engines
 - 3 wagons vs. 6 wagons
- Each wagon will have its own displays, etc
- A new wagon may contain a second MIP

- Part of the OSGi compendium specification
- Event-based communication in OSGi
 - Publishers
 - Subscriber
 - Topics
 - Events

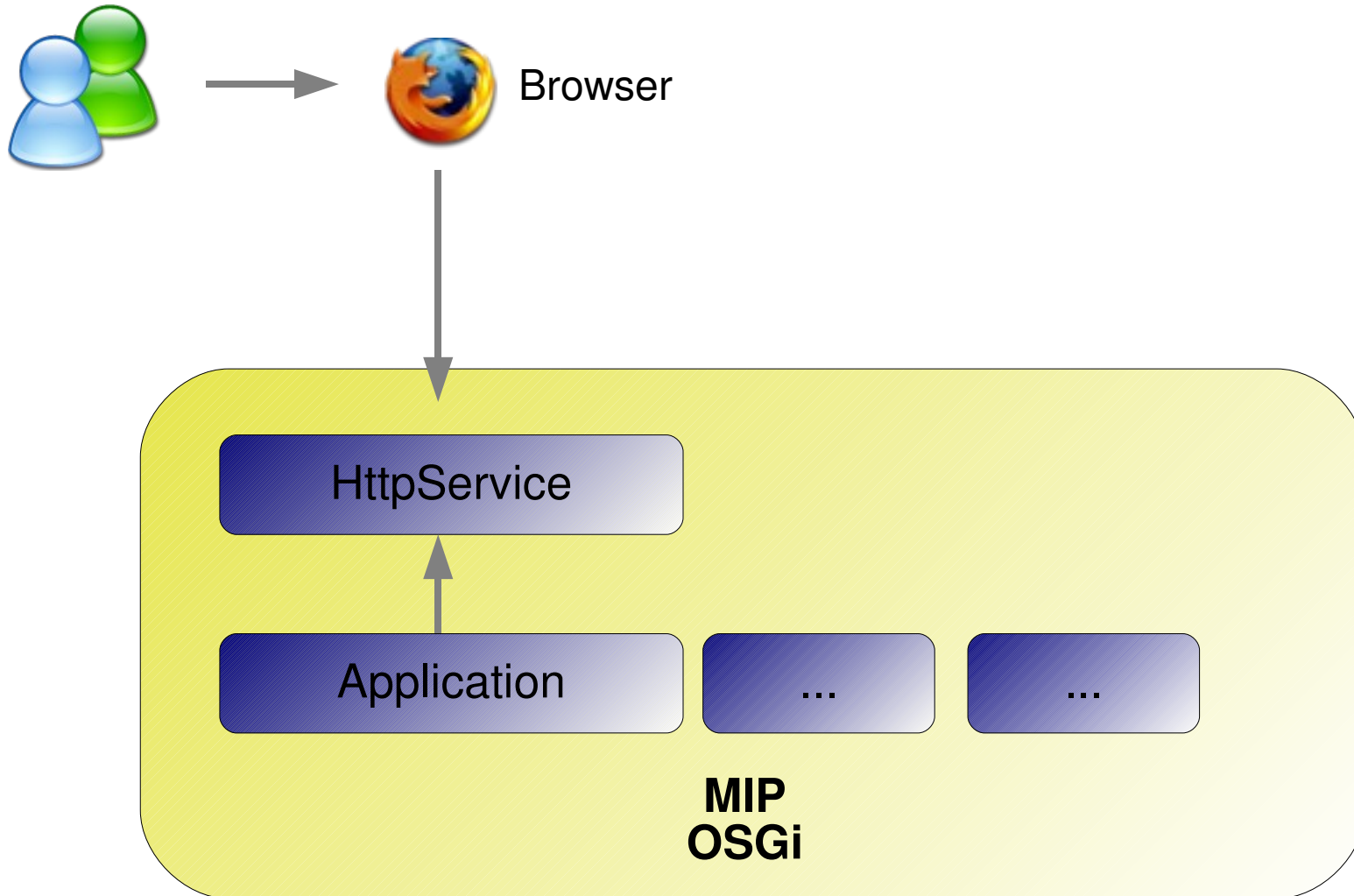
- 90% of the time, bundles communicate with events/topics
- Topic “train/mip/display”
 - Publish events if you have informations that should be displayed
 - Receive events if you are able to display informations
- But...
 - Services decouple consumer/provider
 - Services are extensible (whiteboard pattern)
 - Services have LDAP filters
- So why don't we use services?





- Part of the OSGi compendium specification
- Ability to register HTTP servlets

- Bundles interact with the user by providing a web GUI
 - Apache Wicket, GWT, ...
 - Of course, the underlying JVM has to support the framework



- Marked driven requirements
 - Increasing information needs by mobile customers
 - Cost cutting in operating vehicles
- Technology driven requirements
 - Less integration of applications
 - Stand-alone solutions with individual infrastructure
 - Different life-cycles in IT and vehicles

- Goals for using OSGi framework for on-board integrated solutions in railway vehicles
 - Modularity
 - Scalability
 - Platform independence
 - Extensibility
 - Flexibility