

PARTNER
LOGO

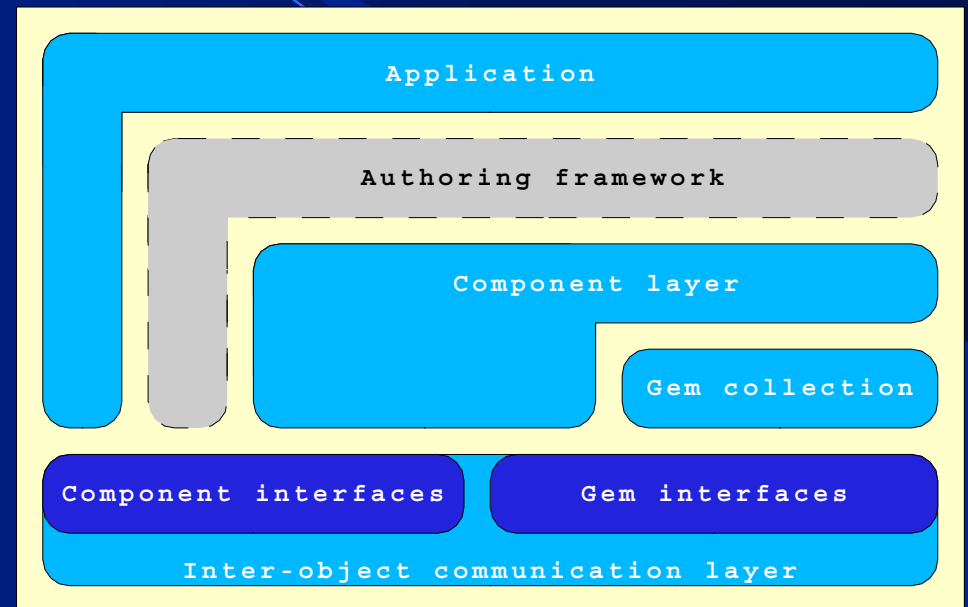
AMIRE Workshop

AMIRE Framework



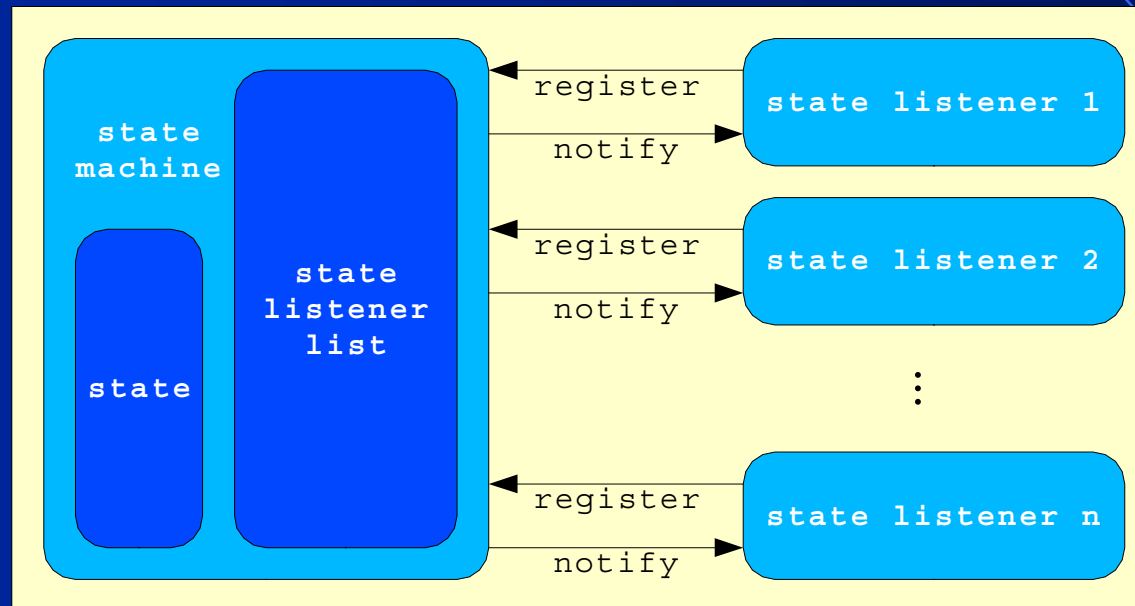
Responsibility

- ❑ Providing basic functionality
- ❑ Integrate gems and components into one similar design
- ❑ Layer abstraction
- ❑ Framework as glue, holding all together



Gem support 1/3

- Communication between gems
 - ✓ Listener concept



Gem support 2/3

- Gem abstraction mechanisms
 - ✓ Classify gems into gem groups by same functionality (e.g. 3d model loader gem group containing a 3ds model loader and a halflife model loader)
 - ✓ Plug-in ability (e.g. image loader or 3d model loader)
 - ✓ Object oriented abstraction
 - ✓ Exchange gem functionality by using other implementations.

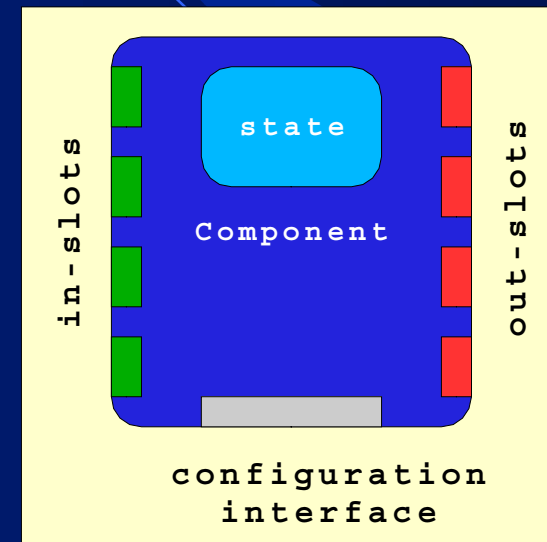
Gem support 3/3

□ Conventions

- ✓ Classify gem groups into gem categories with same behavior (e.g. a loader gem category will contain a 3d model loader and a image loader)
- ✓ Defining nameing and behavior conventions.
- ✓ Ensure a similar design and a good usability (e.g. for a component developer working, who requires the gems)

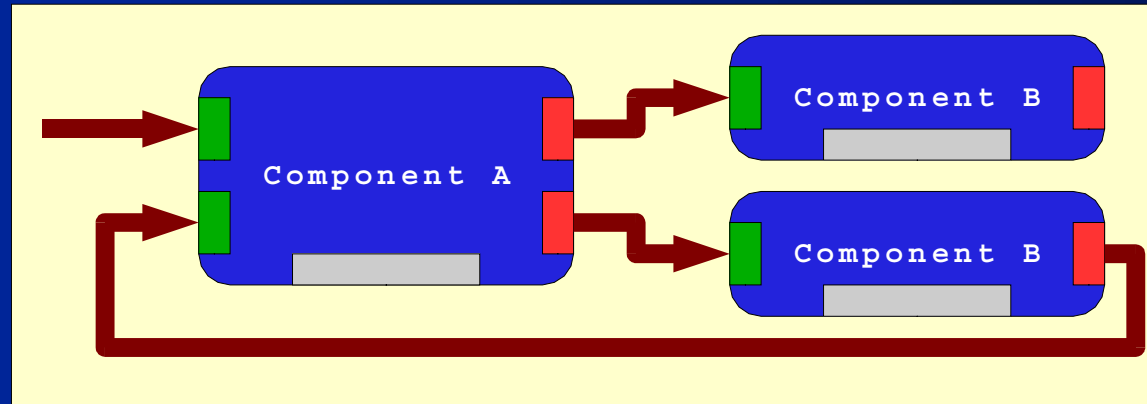
Component support 1/3

- Provide generic interfaces and design
 - ✓ Easy to integrate
 - ✓ Easy to configure
 - ✓ Easy to store and load
 - ✓ State oriented



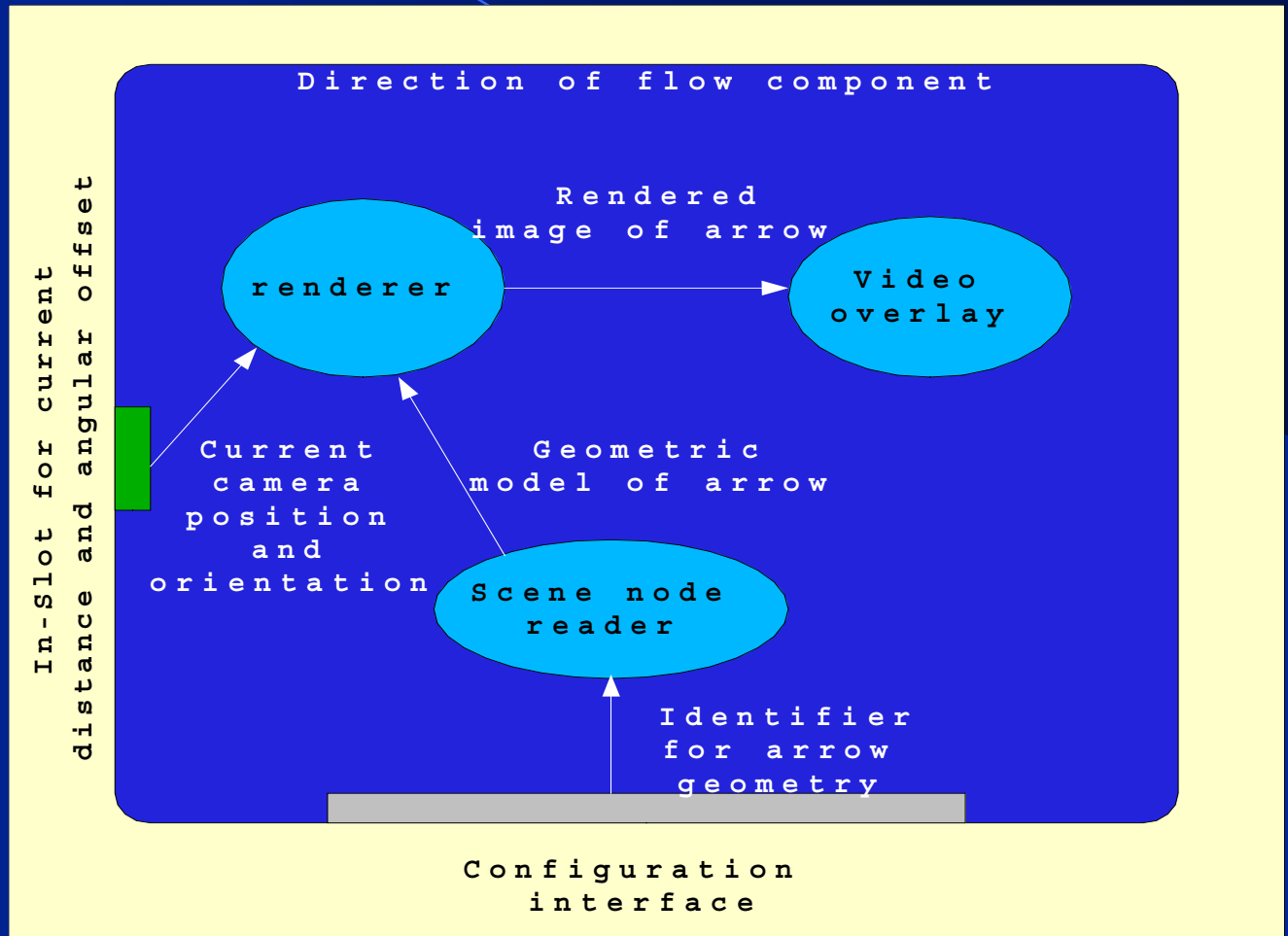
Component support 2/3

- Connections between components
 - ✓ Create connections => component network
 - ✓ Store and load connections
 - ✓ Support authoring tool by generic interfaces



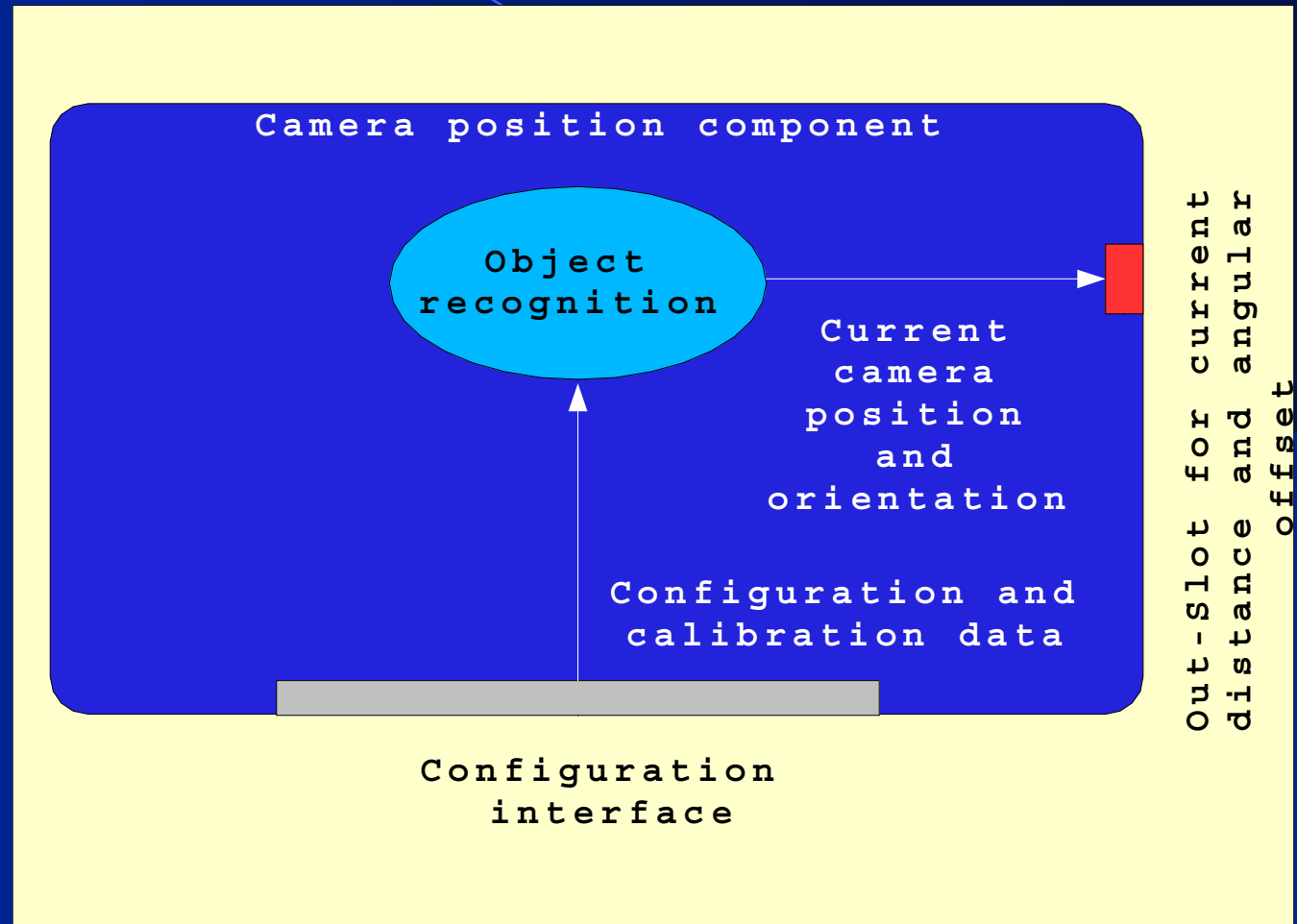
Component example 1/3

- ❑ Consists of gems
- ❑ Configurable geometry
- ❑ Camera position as input
- ❑ Produces overlaid image

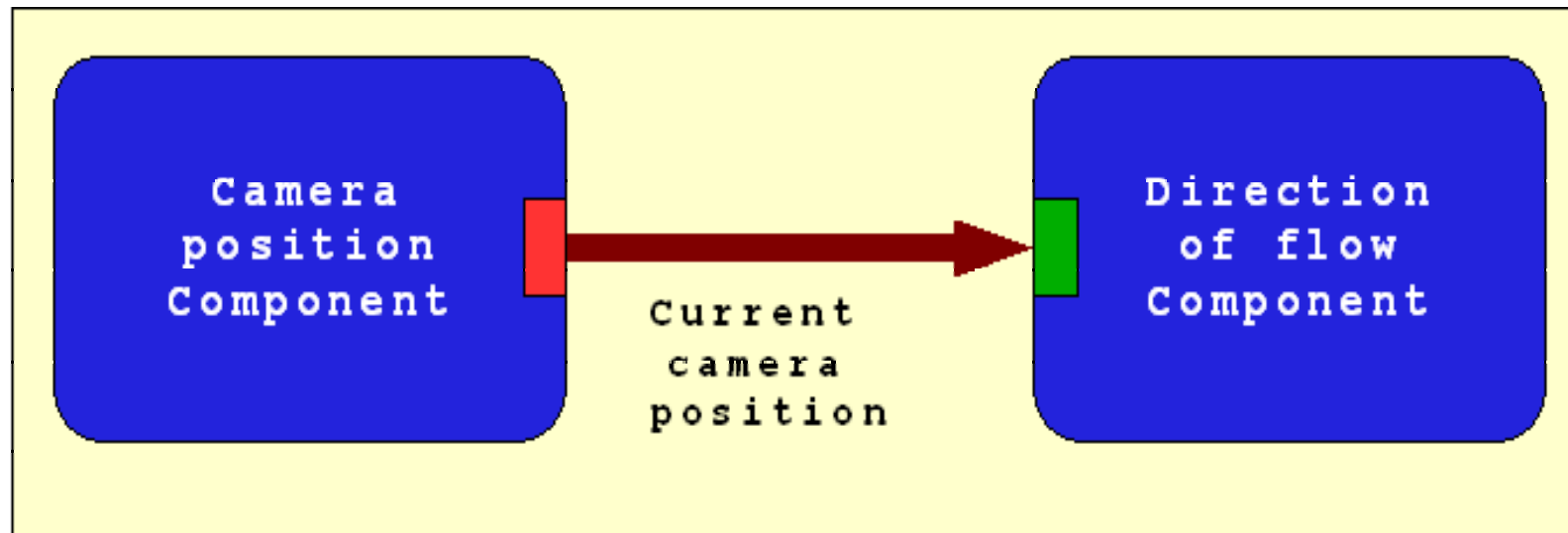


Component example 2/3

- ❑ Configurable object recognition
- ❑ Provides the camera position as output



Component example 3/3



Cooperations

- ❑ Gem developers (C-LAB): classify the gem groups and define conventions
- ❑ Component developers (LABEIN): provide and maintain the generic component interfaces and base classes
- ❑ Authoring developer (FHG): provide and maintain generic access to the components