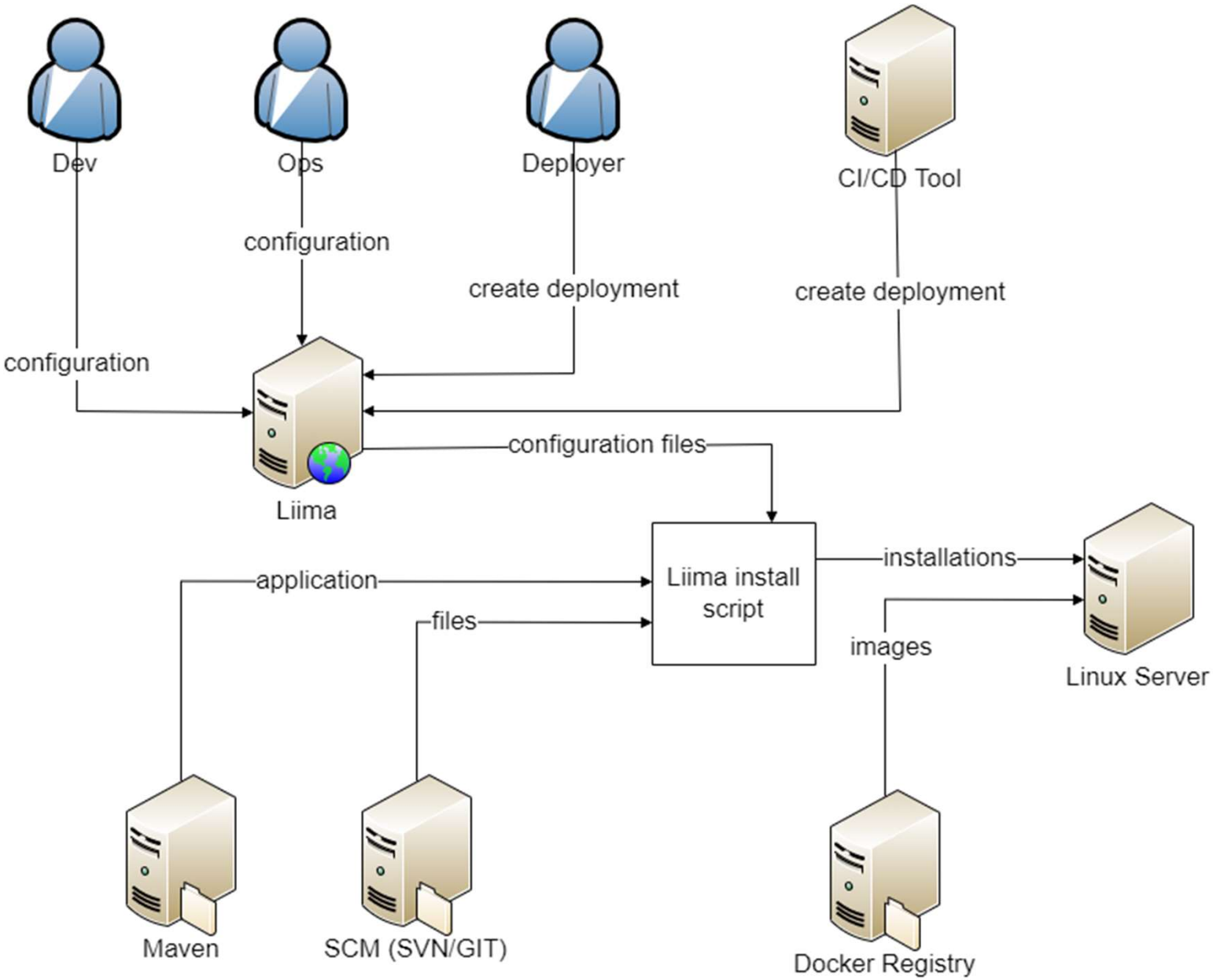
Liima Features

What is Liima? What are it's advantages?

- **Liima** (Finnish for glue) manages the configuration and deployment of applications. It's optimized for configuration reuse.

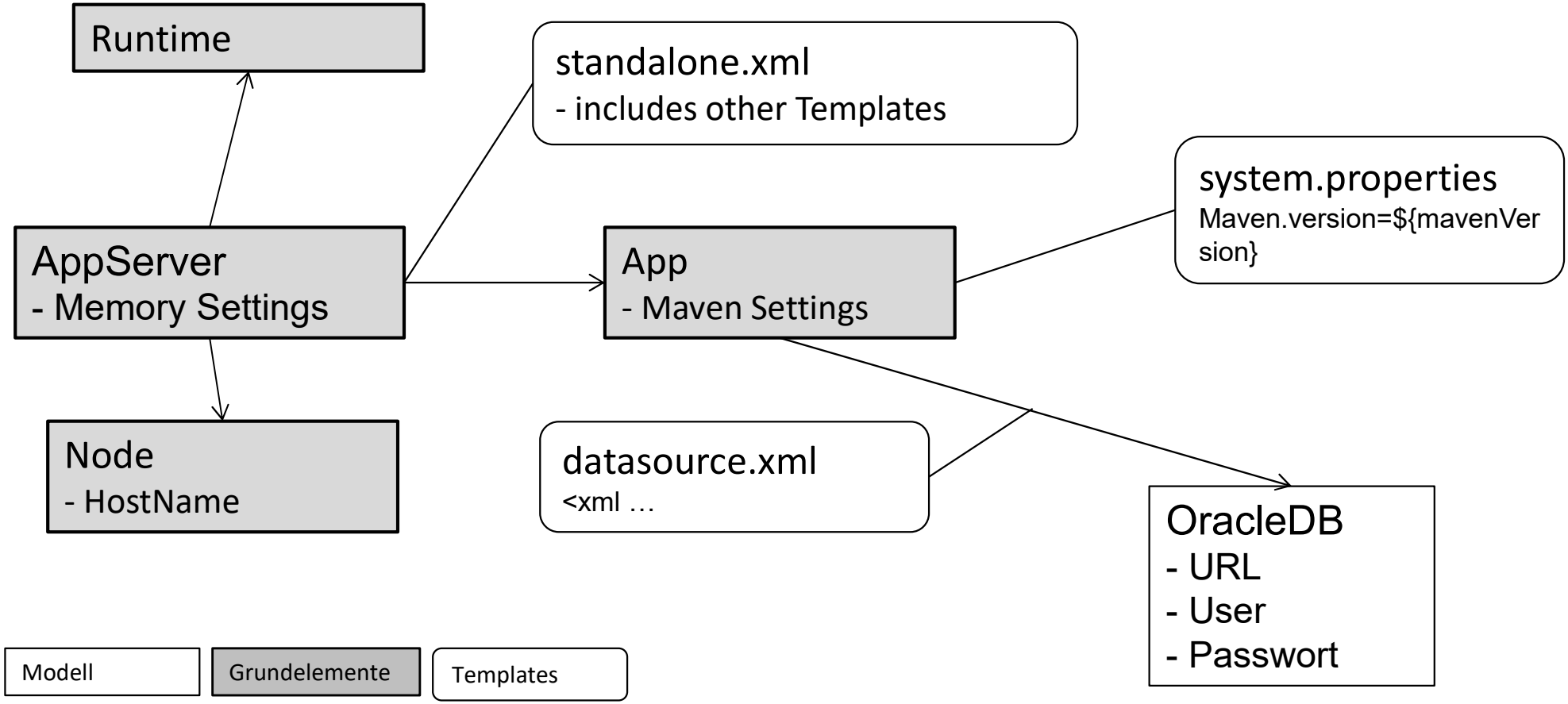
- Main advantages of Liima
  - Configuration doesn't depend on the technology of the application
  - Avoids duplication of configuration
  - Versioning and audit
  - The configuration is installed with the configuration as one package
  - Configuration is readable by different teams, secured by role base access control
  - Easy to integrate with other tools and technologies

Deployment Prozess mit Liima



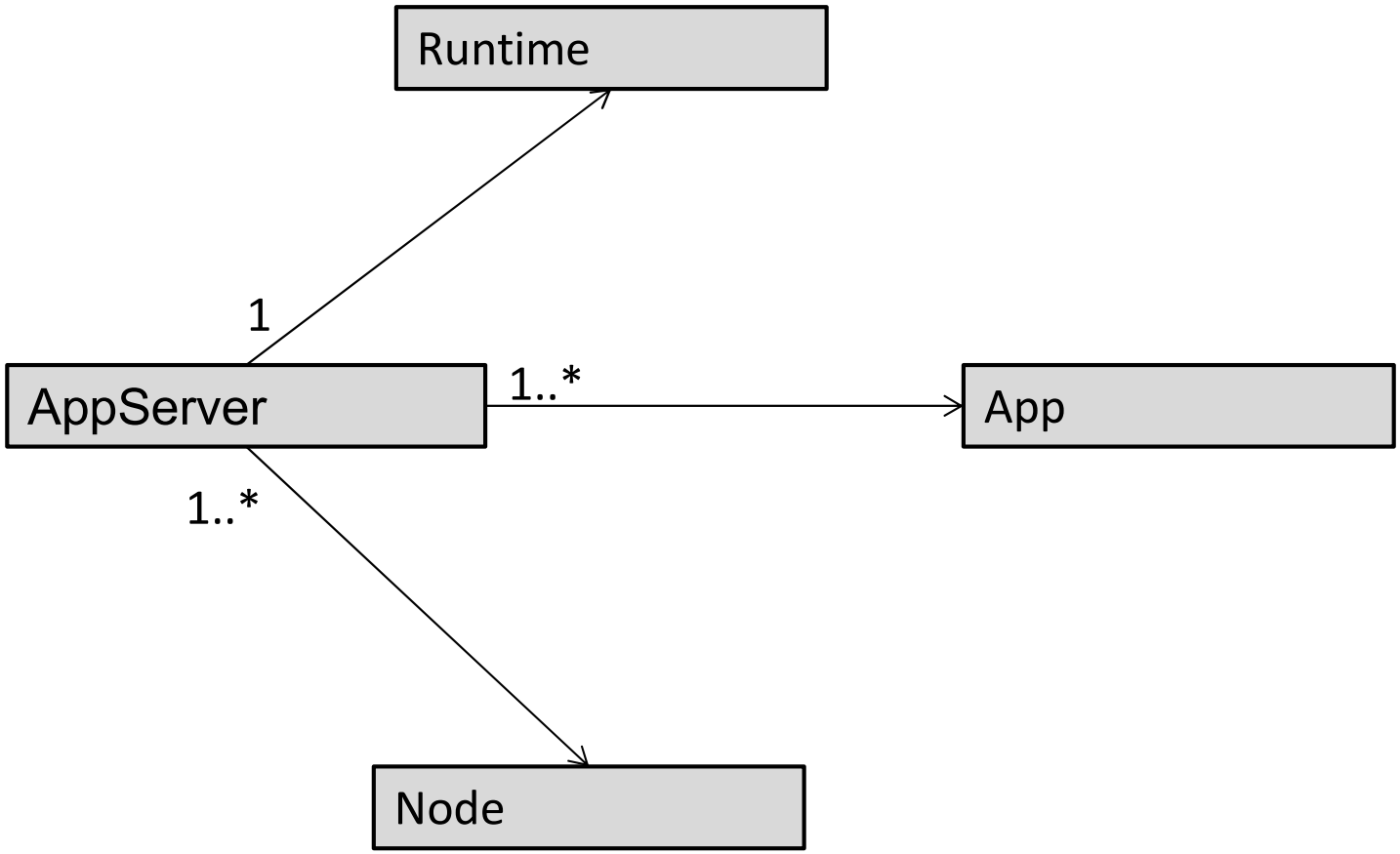
Configuration model

- Freely definable object model with Liima resources
- Modelling of applications with their relations
- Freemarker templates convert the model into configuration files



Modell: default resources

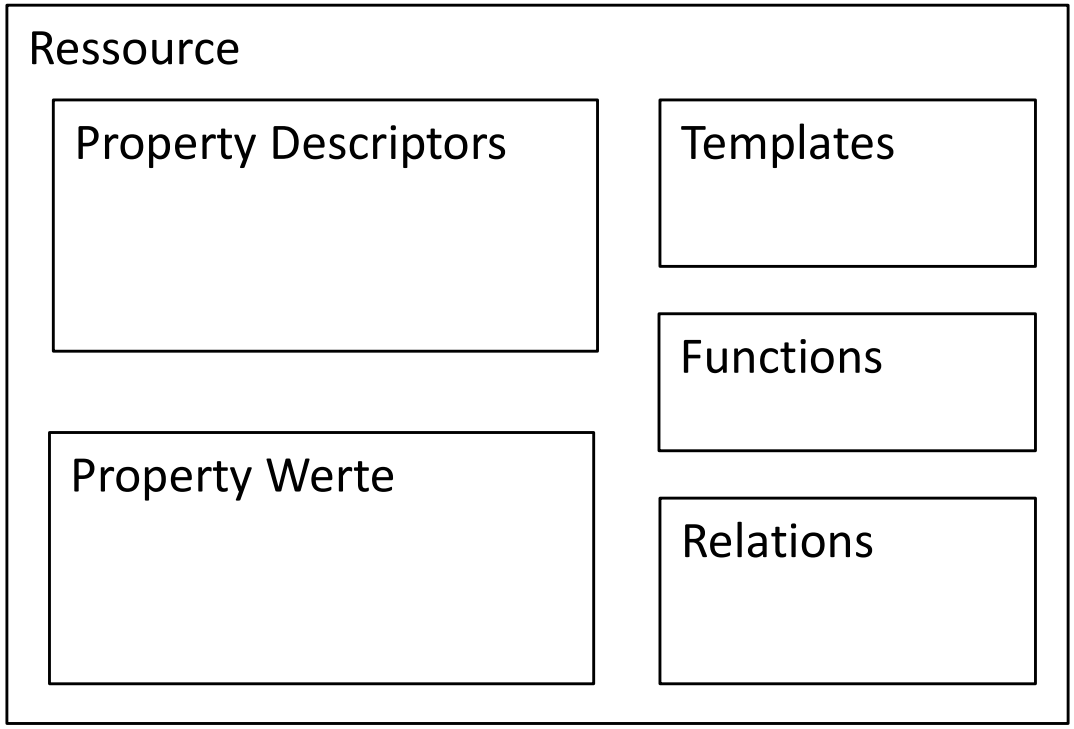
- Default resources are required for each deployment



Modell | Grundelemente

Model: resources

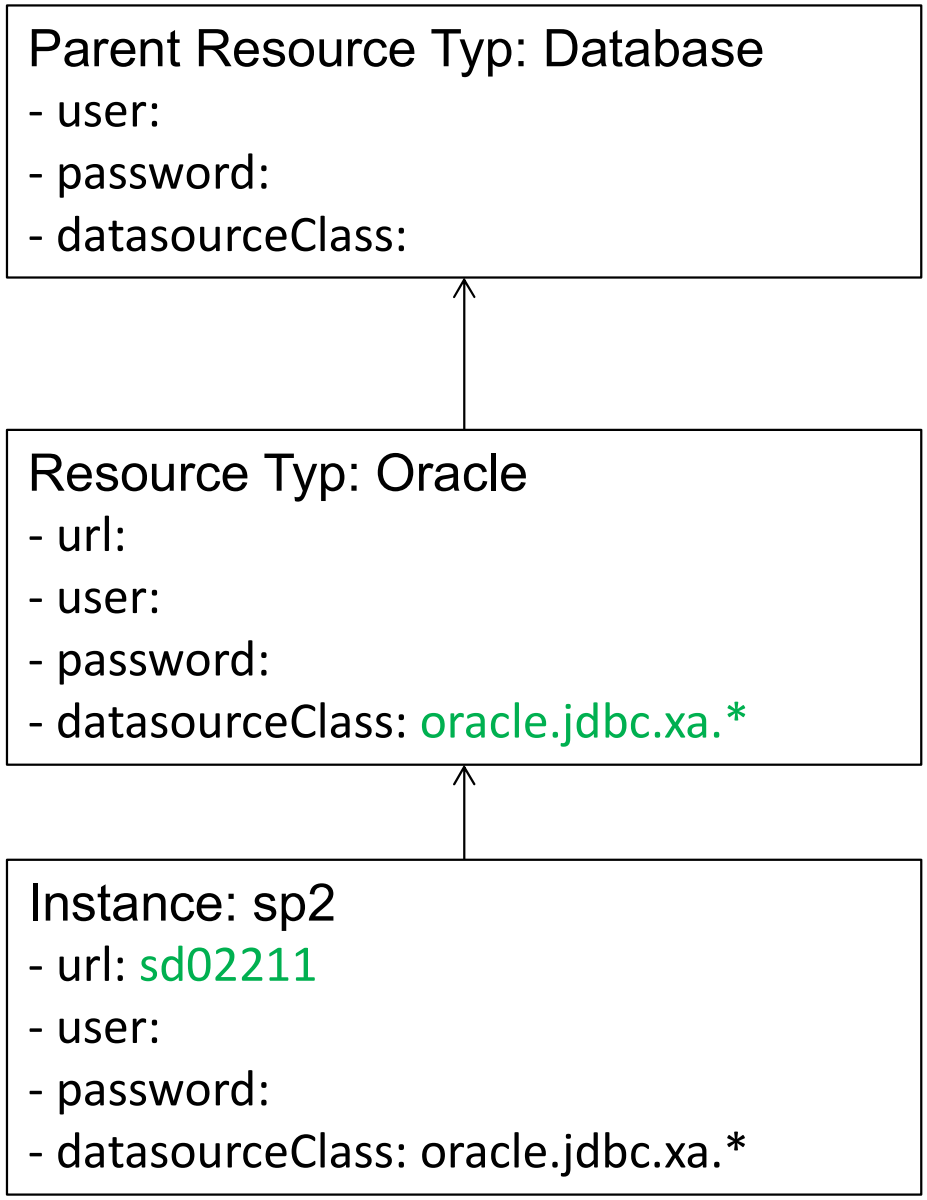
- Resources contain:
  - Properties descriptors: definition of a property
  - Property values: value of a property
  - Templates: convert the properties into configuration
  - Functions: composition of properties
  - Relations: reference to other resources for reuse



Model: inheritance and properties

- Every resource has a type
- A type can have one parent type
- Resources of a type are called instances
- The following will be inherited:
  - property descriptors
  - property values
  - templates
  - functions

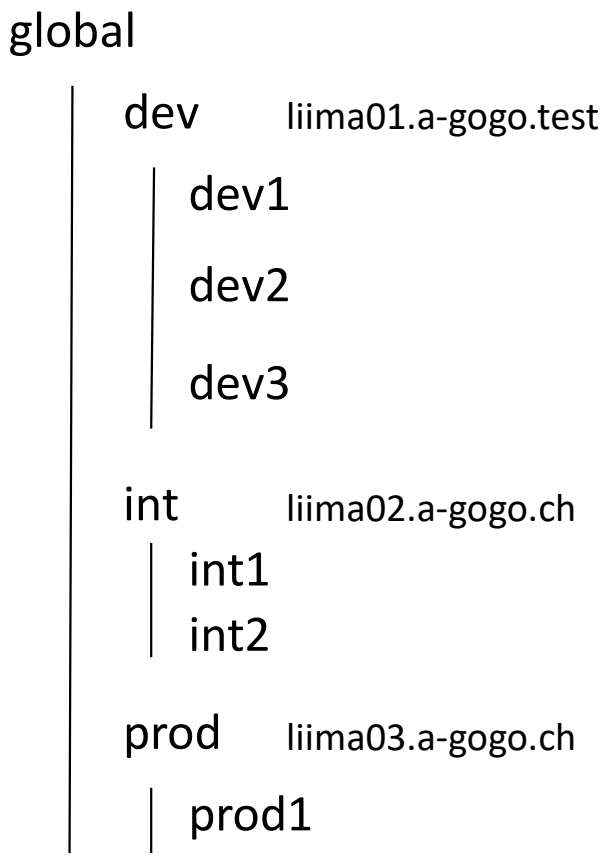
- Green: defined on that resource
- Black: defined on a parent resource type



Model: environment hirarchy

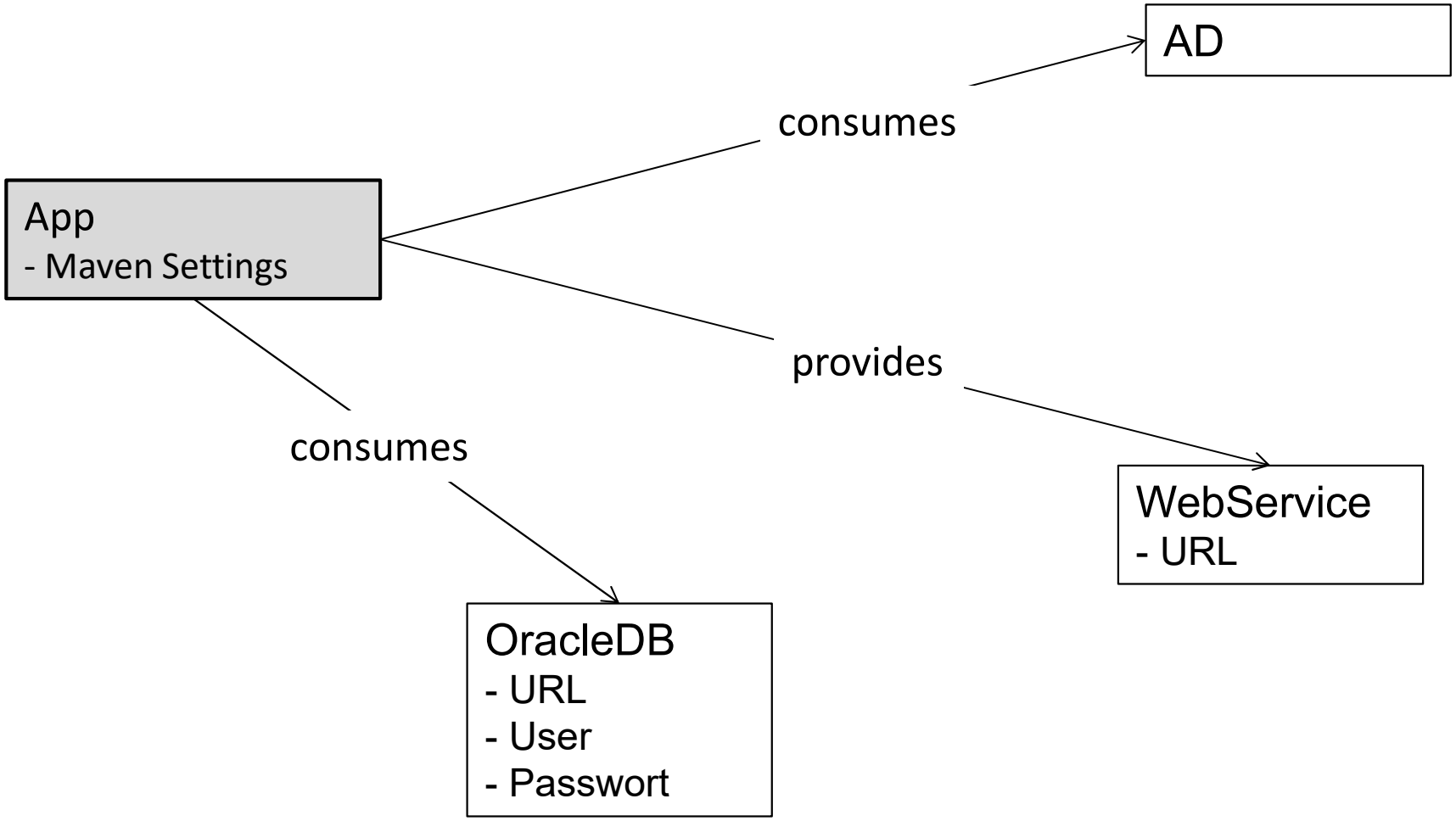
- Property values are inherited from global to domain (dev, int, prod) to environment
- Enables defaulting
- Avoids redundancies

- On the Liima UI:
  - Green: defined on this level
  - Black: inherited from a higher level
  - Red: validation error
  - The tooltip (i) shows where a inherited property was overwritten



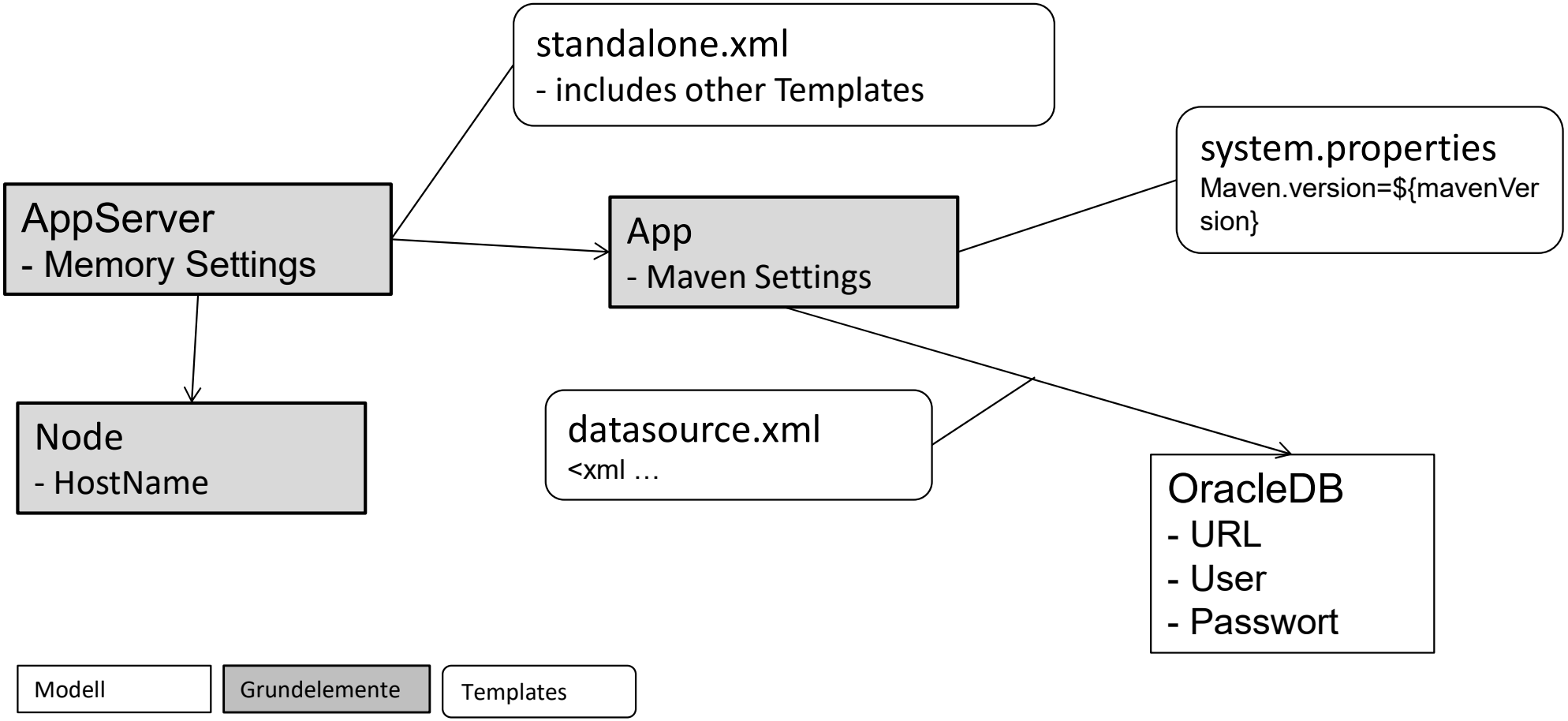
Model: relations

- Consumed: a foreign resource is consumed
- Provided: a resource is provided
- Templates can access related resources
- Property values can be overwritten in relations



Model: templates

- Templates are use to write configuration
- Uses Freemarker: http://freemarker.org/docs/
- Templates can be attached to resources, resource types and relations
- Templates are always connected to a runtime



Releasing: motivation

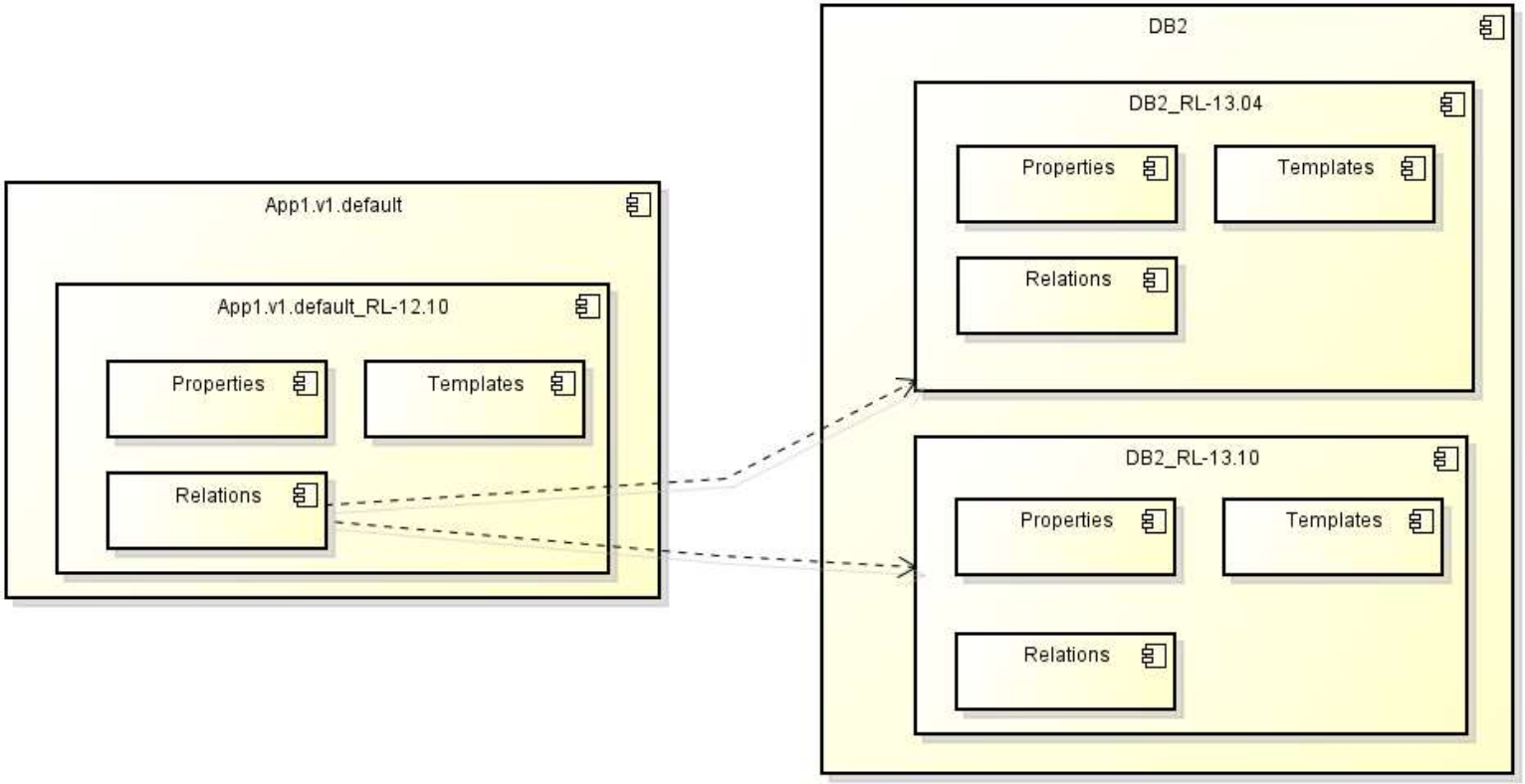
- A Liima configuration is always valid for all environments
- Releasing enables to have different configurations on different environment

- Example:
  - RL-15.04 was released recently, next release is RL-15.10
  - An application need a different database in RL-15.04
  - The existing config can't be modified because it would break productions deployments which still need the old database

- Workarounds:
  - Before each deployment change the configuration by hand
  - Templates with #if and #else
  - Different Liima configuration per environement

Releasing

- A release is a copy of a resource: properties, templates, relations etc.
- Different release are shown in Liima as one element
- Corresponds to releases of the application
- Is similar to a branch in source management
- Relations of consumers are extended automatically
- At deploy time the matching relation will be selected

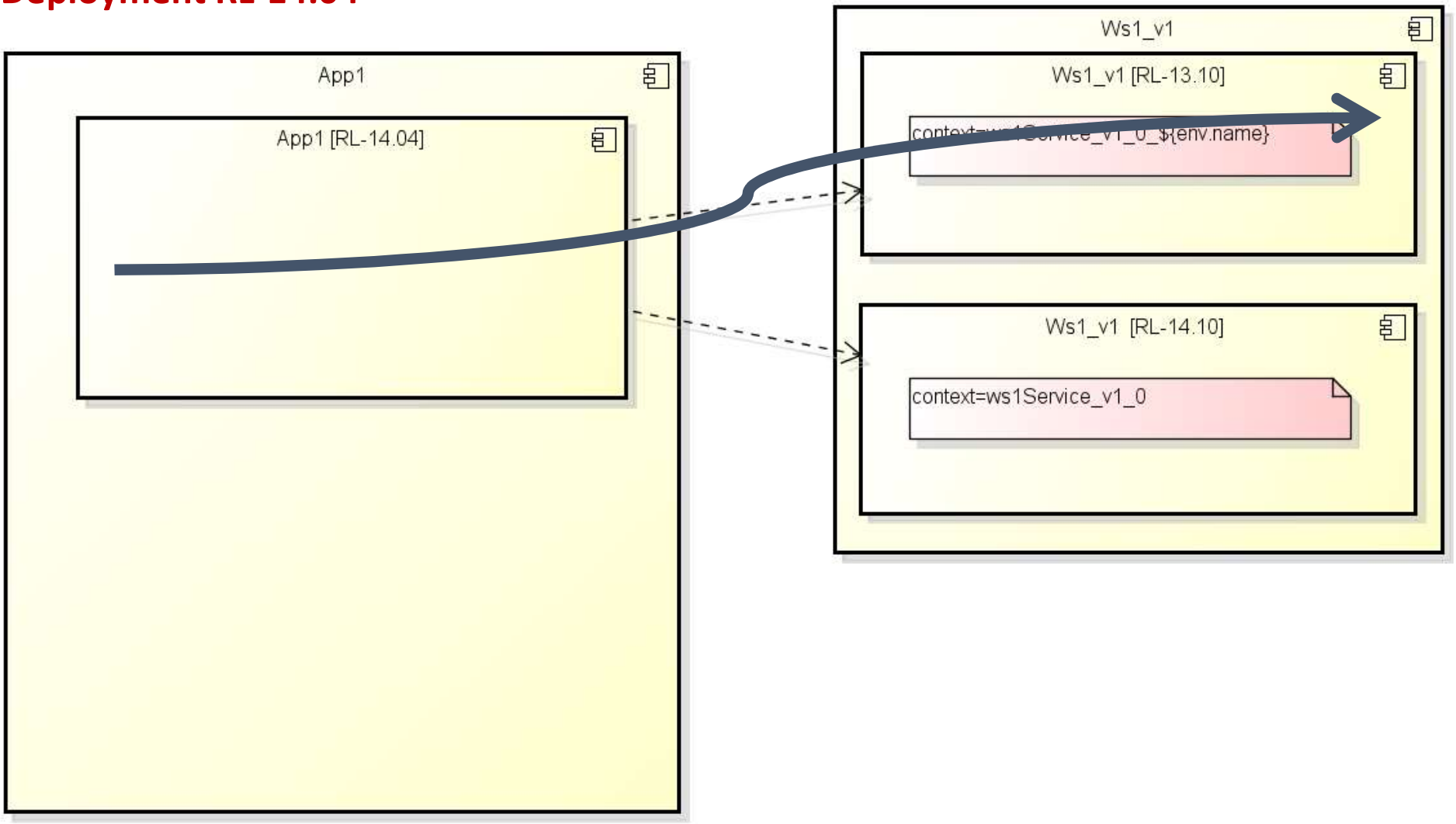


Releasing: deployment

- The release dropdown in the deployment UI determines which resource releases are selected
- The releases in the dropdown depend on the releases of the app server
- Applications can deployed with existing releases of an app server or higher

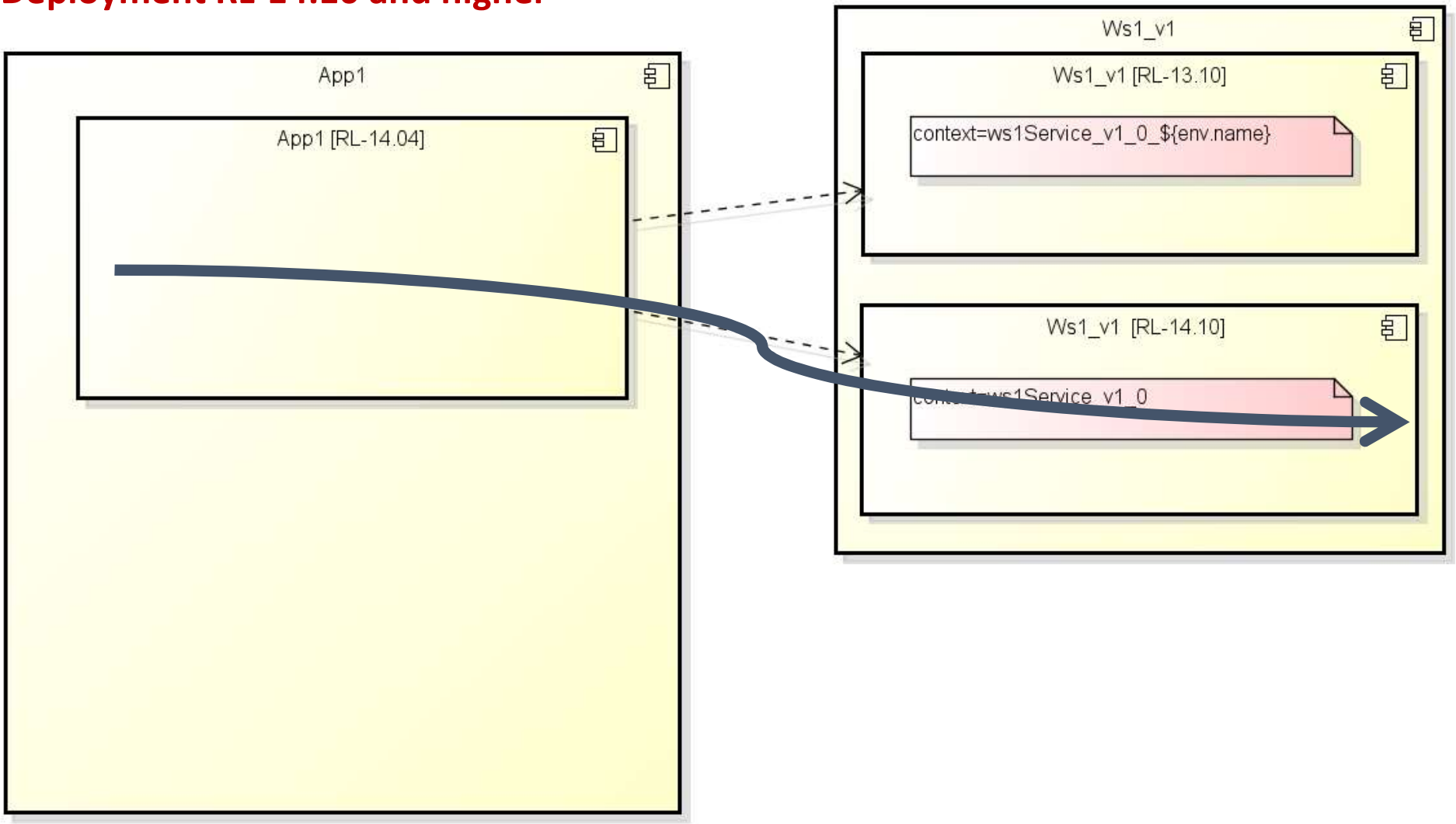
Model: releases

Deployment RL-14.04



Model: releases

Deployment RL-14.10 and higher



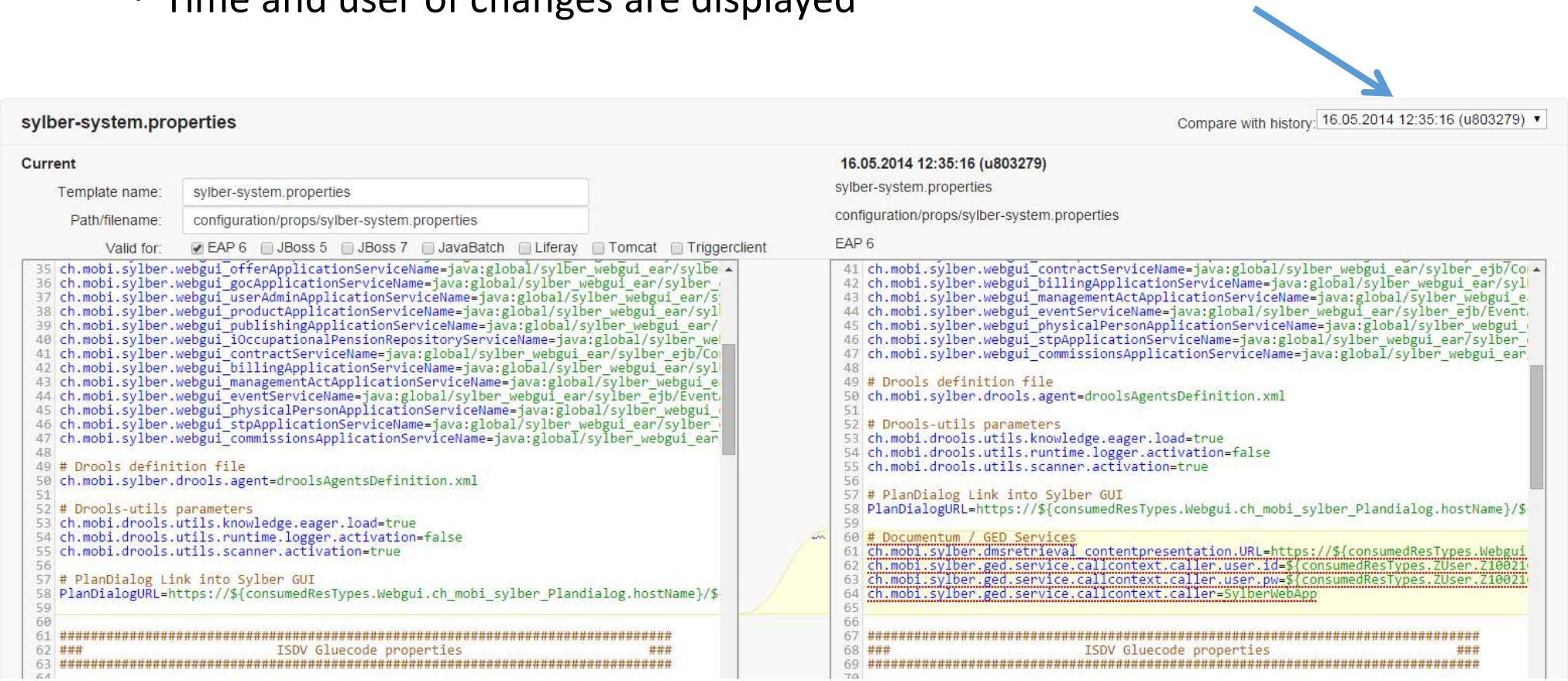
Template editor

- JavaScript based template editor (CodeMirror)
  - Built in search function with highlighting
  - Freemarker syntax highlighting
  - Row numbers
  - Fullscreen mode
  - Most import key shortcuts blow the editor window

- Freemarker syntax is check when a template is saved
  - Validates unclosed bracket and tags
  - Can't check if variables are missing! Use test generate for this

Template editor: compareision

- A template can be compared with older versions
  - Editor shows differences
  - Supports merging
  - Time and user of changes are displayed



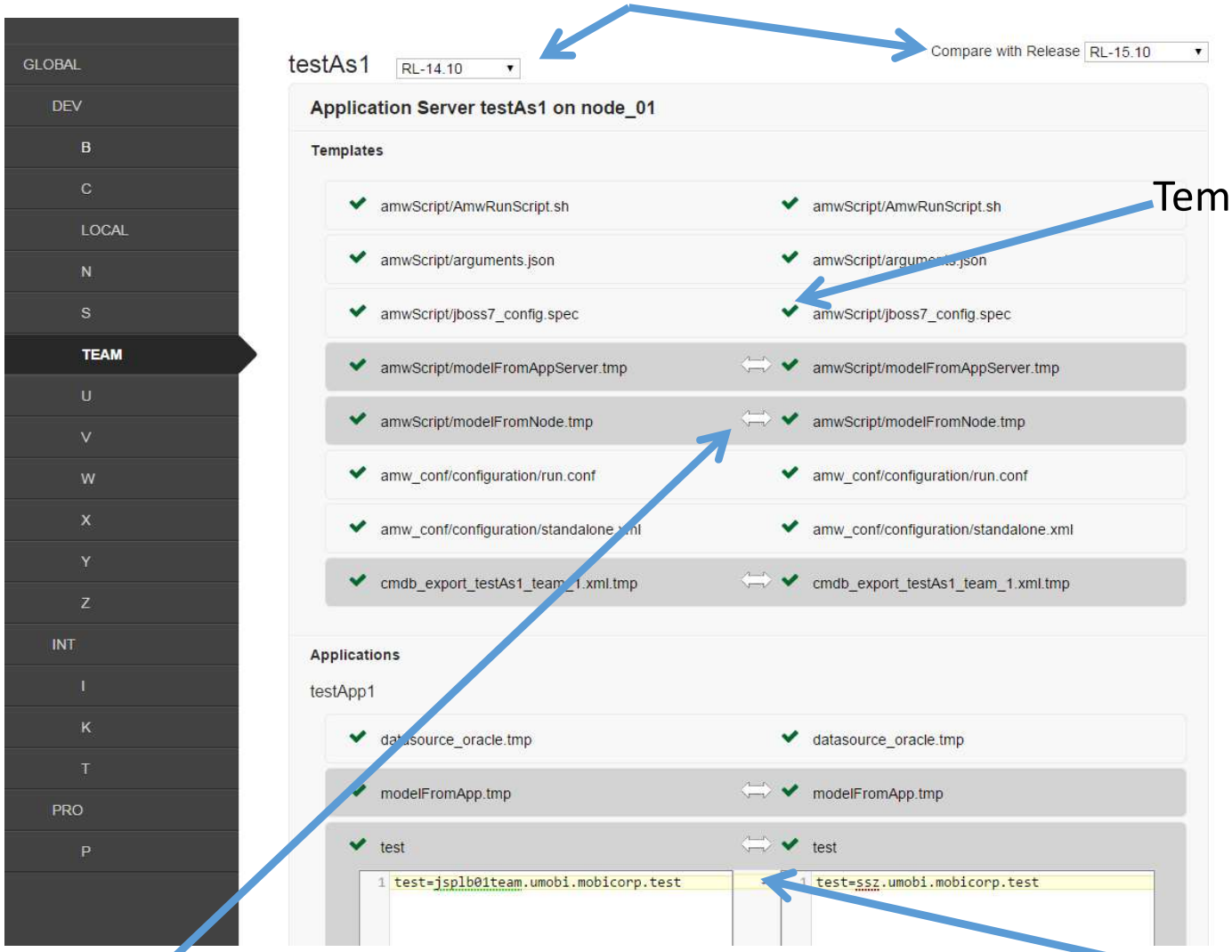
Test generate

- Function reachable via „Test Generation“ button on app or app server
- Show rendering errors in templates
- Shows the templates after rendering
  - Is only shown if the user has permissions to deploy to that environment.

- Releases can be compared to each other
  - Uses the diff function of the editor
  - Can also be used to compare configurations from specific dates

Test generate

Release Selektion



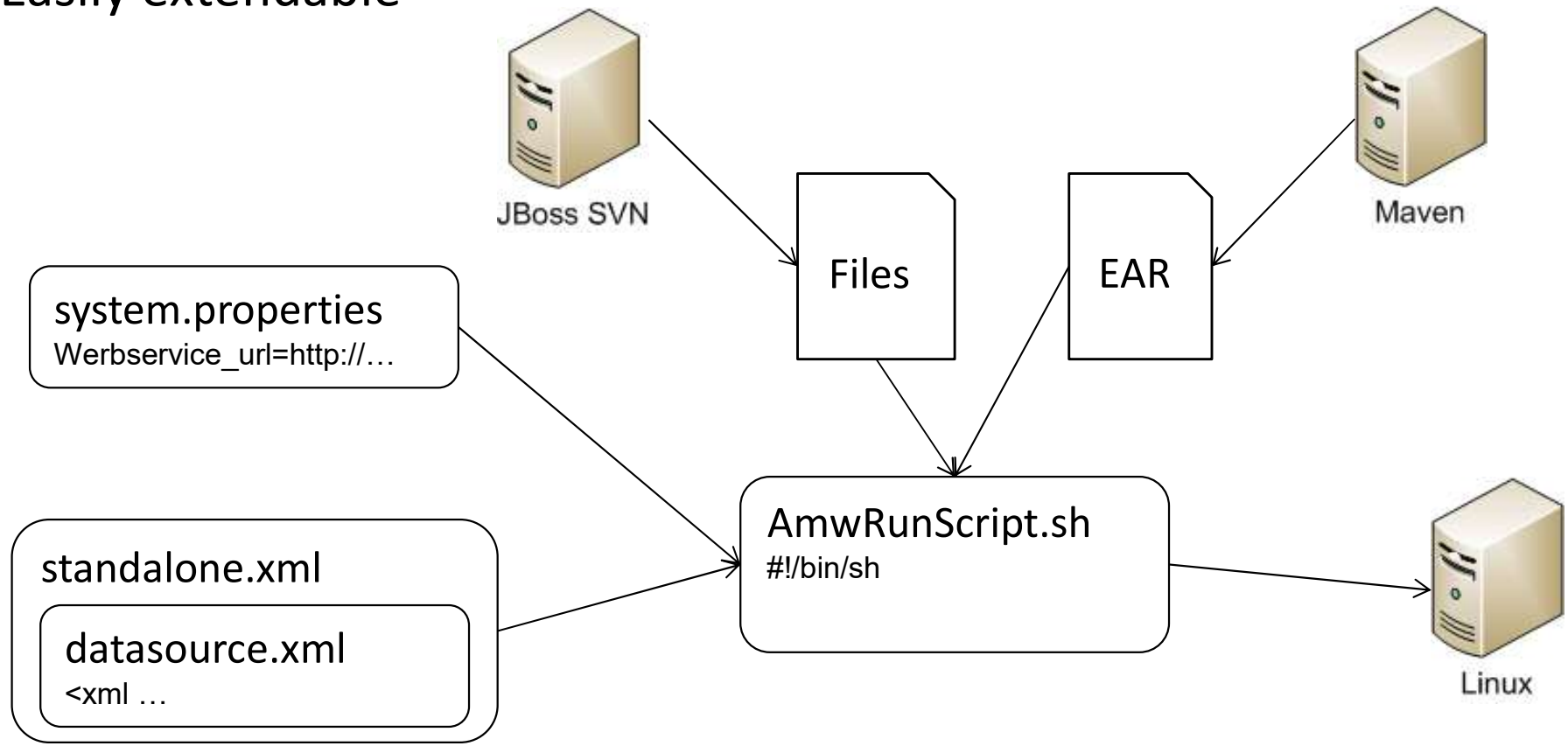
Templates validation

Are there differences?

Template differences

Deployment

- Packaging of configuration and application
- Installation on the target system
- Easily extendable



Liima Templates

Deployment filter

- Filter are connected automatically by ANS or OR depending on the filter
- Deployment can be exported as CSV
- Filter "Latest deployment job for App Server and Env": shows only the latest deployment for one environment

Best practices

- Avoid redundancies
  - Enables central management
  - Better overview
  - Properties can contain other properties: ${env.name?lower_case}
- Use resources instead of properties
  - Easier reuse
  - Dependencies are visible easier
- Fill out all environments from the beginning
  - Less prone to forget configuration
  - Less errors