

Getting Started with QCMagritte

Diego Lont & Stephan Eggermont

- Download a QCMagritte image from <http://ci.inria.org/pharo-contribution/QCMagritte>
- Download a pharo vm from <http://get.pharo.org>
- Download a QCMagritte demo from <http://smalltalkhub.com/mc/DiegoLont/QCMagritteDemo>

What is QC-Magritte

- Application level framework
 - On top of Seaside and Magritte
 - Provides re-usable parts for common tasks

QCMagritte supports ...

Application
Template

Commands

Application
navigation

Table support

User
management

Hierarchical
domain
structure

Multi-
language

Search
(Queries)

Influences
(AJAX)

Menu
structure

Example: CI

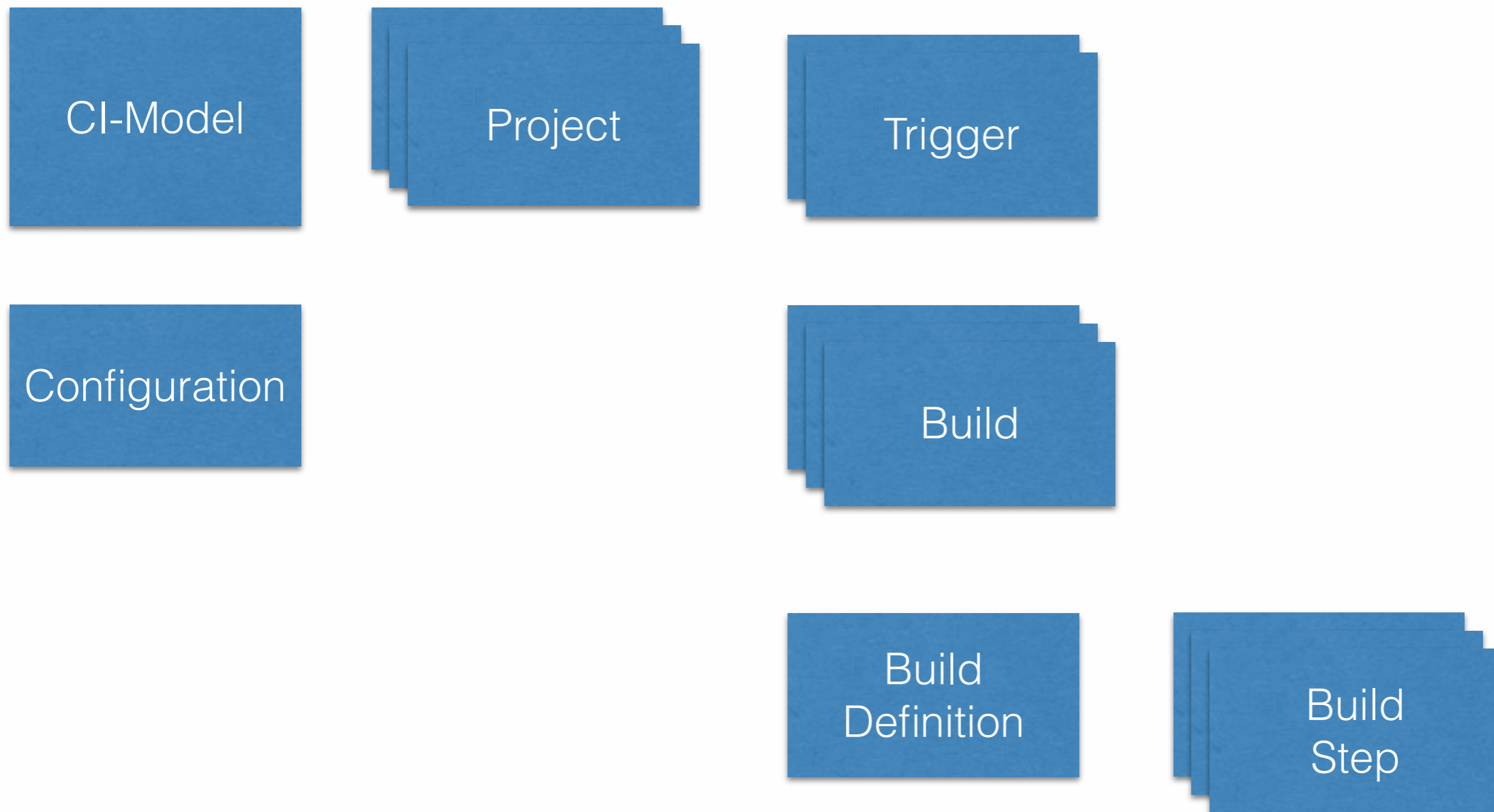
- Continuous Integration Server

The screenshot shows the Jenkins web interface for the 'Contribution' section. It features a sidebar with navigation links, a main header with the 'Contribution' title, and a table of projects. The table columns include 'Name', 'Number of builds', and 'Last Success'. The projects listed include various Android-related builds and tools.

Name	Number of builds	Last Success
Android-Cli-OS-Tools	1	2 hrs 10 mins - #147
Android	3	1 yr 3 hrs - #120
ARM-Android	1	2 hrs 1 day - #250
ARM-Android-Debian	9	N/A
ARM-Android-OpenGLES	9	23 days - #7
ARM-Android-LocalDebian	9	2 hrs 0 days - #10
Android	1	19 hrs - #300
android	1	4 hrs 3 days - #3
Android	2	14 hrs - #260
Android	2	20 hrs - #340
Android	2	20 hrs - #360
Android	2	7 hrs 36 mins - #380
Android	2	9 hrs 22 mins - #400

- Not focus of the talk
- Predefined with Magritte definitions

CI Model



Demo

- View magritte descriptions in model

Template

- Application
 - Seaside application, registering itself
 - Points to model
 - Creates menu
- Model
 - Describes entry points for entire domain
 - Defines builders used to generate components

Bootstrap template

```
QCBootstrapApplication subclass: #CIBuildServerApplication
```

```
CIBuildServerApplication>>title  
  ^'Demo CI'
```

```
CIBuildServerApplication>>model  
  ^CIBuildServerModel default
```

```
CIBuildServerApplication class>>applicationName  
  ^'Demo CI'
```

```
QCBootstrapApplicationModel subclass: #CIBuildServerModel
```


Demo

- Show code
- Run tutorial:1
- Show application
- Add 2 projects
 - 1 project defines steps and triggers

Default commands

- CRUD
- User rights
- Domain constraints

```
CIProject>>canRemove  
  ^self builds isEmpty
```

Custom commands

- Domain object has containerActions
 - Label
 - Condition
 - Action

```
CIProject>>containerActions: aContainer  
<magritteContainer>  
^aContainer  
  addCommand: 'build' condition: #canBuild action: #startBuild;  
  yourself
```

Custom components

- Easy extend model with custom components
- Demo: add status description + component

User Management

- Only allow changes by logged in users
- Restrict access to views/actions
- Controlled by “hasUserManagent”
 - Defaults to true if there is an admin user
- (CRUD) Rights and roles model.

Demo

- Add user management

QCMagritte supports ...

Application
Template

Commands

Application
navigation

Table
Support

User
management

Hierarchical
domain
structure

Multi-
language

Search
(Queries)

Influences
(AJAX)

Menu
structure

Influences

- disable timeout
- user first, last name
- How to avoid cycles

Application navigation

- Mostly announcement based (still some call's left, slowly being replaced)
- QCPageChoice shows how

Parent Object

- Hierarchical ownership model for the domain
- Only one owner: parent
- Backlink to model = parent model

- QCQueriedToManyComponent
 - search support
 - newInstance sets parent if needed

Multilanguage

- In-application support for multiple languages
- Each user can have own language preference
- Application has default language
- Keep default values for untranslated
- Default translation of labels, groups, comments, error messages. Others: `asMultilanguageString`

Search

- Model-wide search, from a starting point
- Description-type specific
 - everything transformed to string representation

Table support

- Select on field values, smart filter
- Sorting
- Export CSV
- Select columns
- Add calculated columns

Menu structure

- Tree based, announcements
- Autocollapse: expand current selection