

J2J

JAVAFX TESTING WITH JUBULA

by @MarkusTiede - BREDEX GmbH

AGENDA

What's Jubula?

The concepts...

What's working?

Having some fun...

Some insights...

Getting started...

WHAT'S JUBULA?



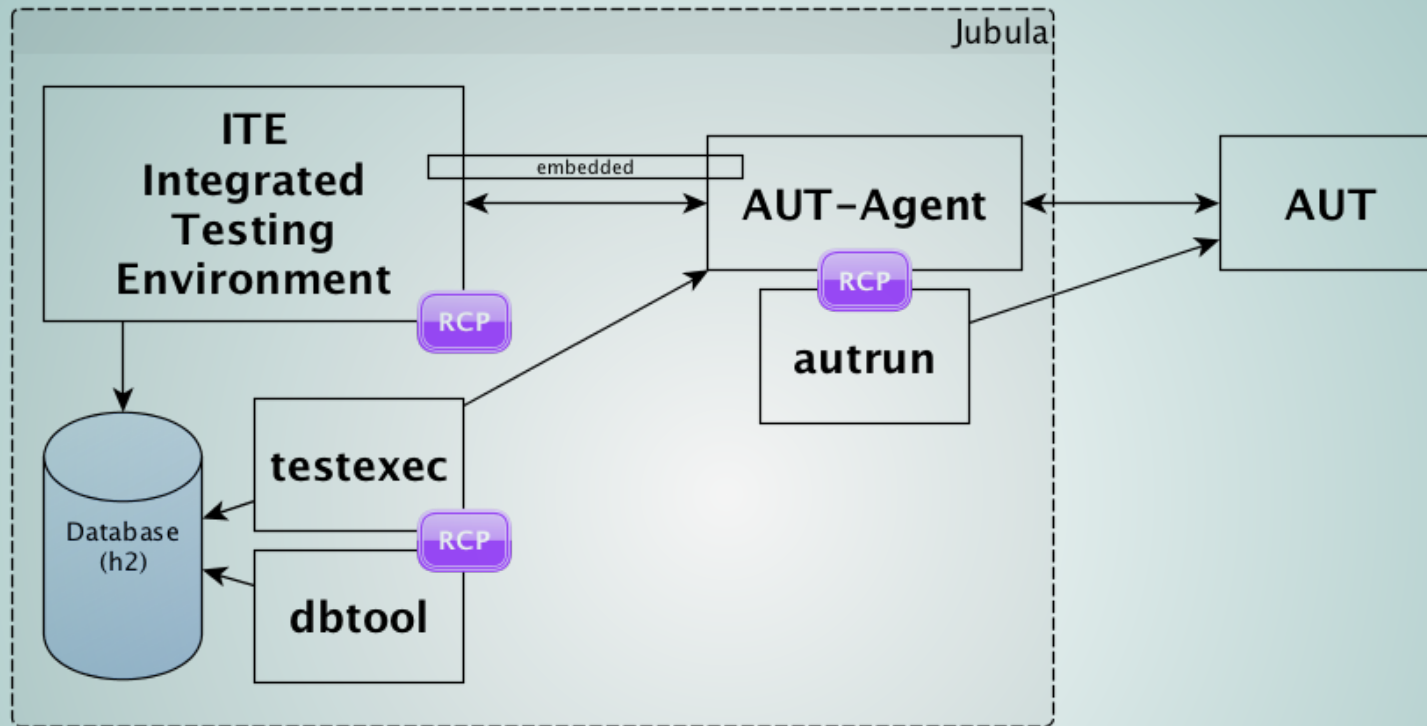
@eclipse.org/jubula since 2011

automated functional GUI testing

Code-free keyword-based black-box approach

human-readable test steps + robust UI widget recognition

JUBULA - THE BIG PICTURE

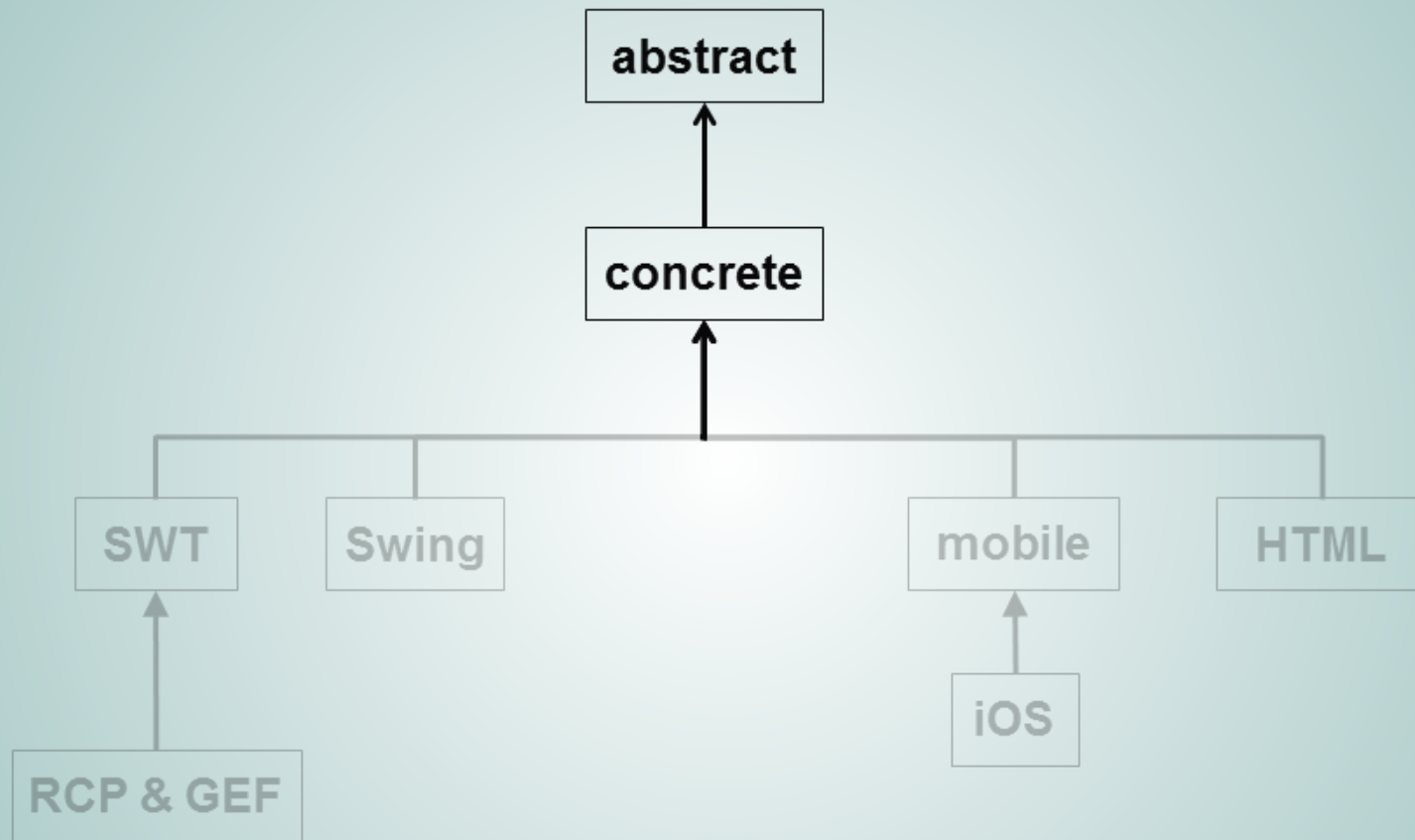


AUT: **A**pplication **u**nder **T**est

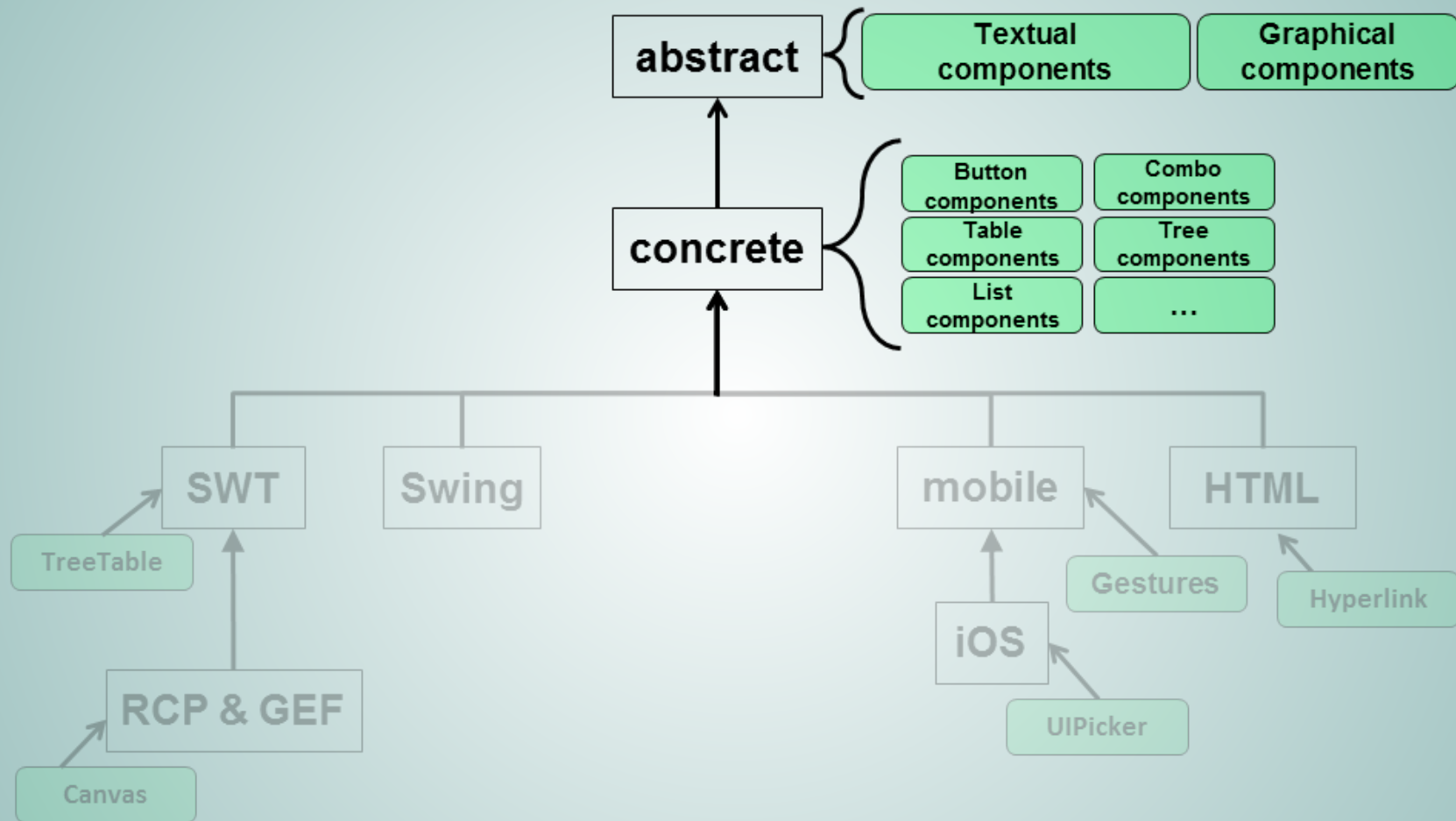
THE CONCEPT: TOOLKITS



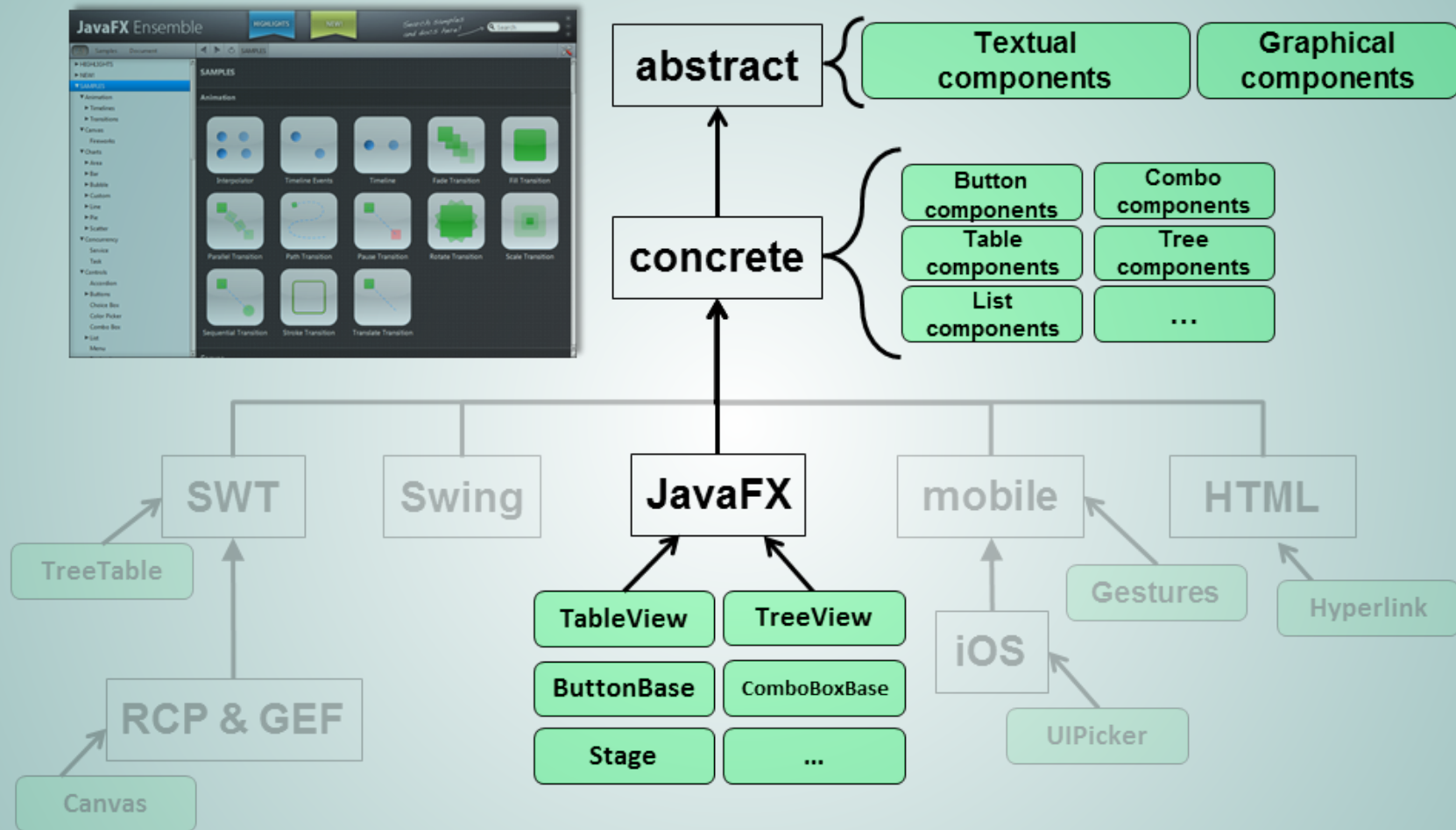
THE CONCEPT: TOOLKIT ABSTRACTION



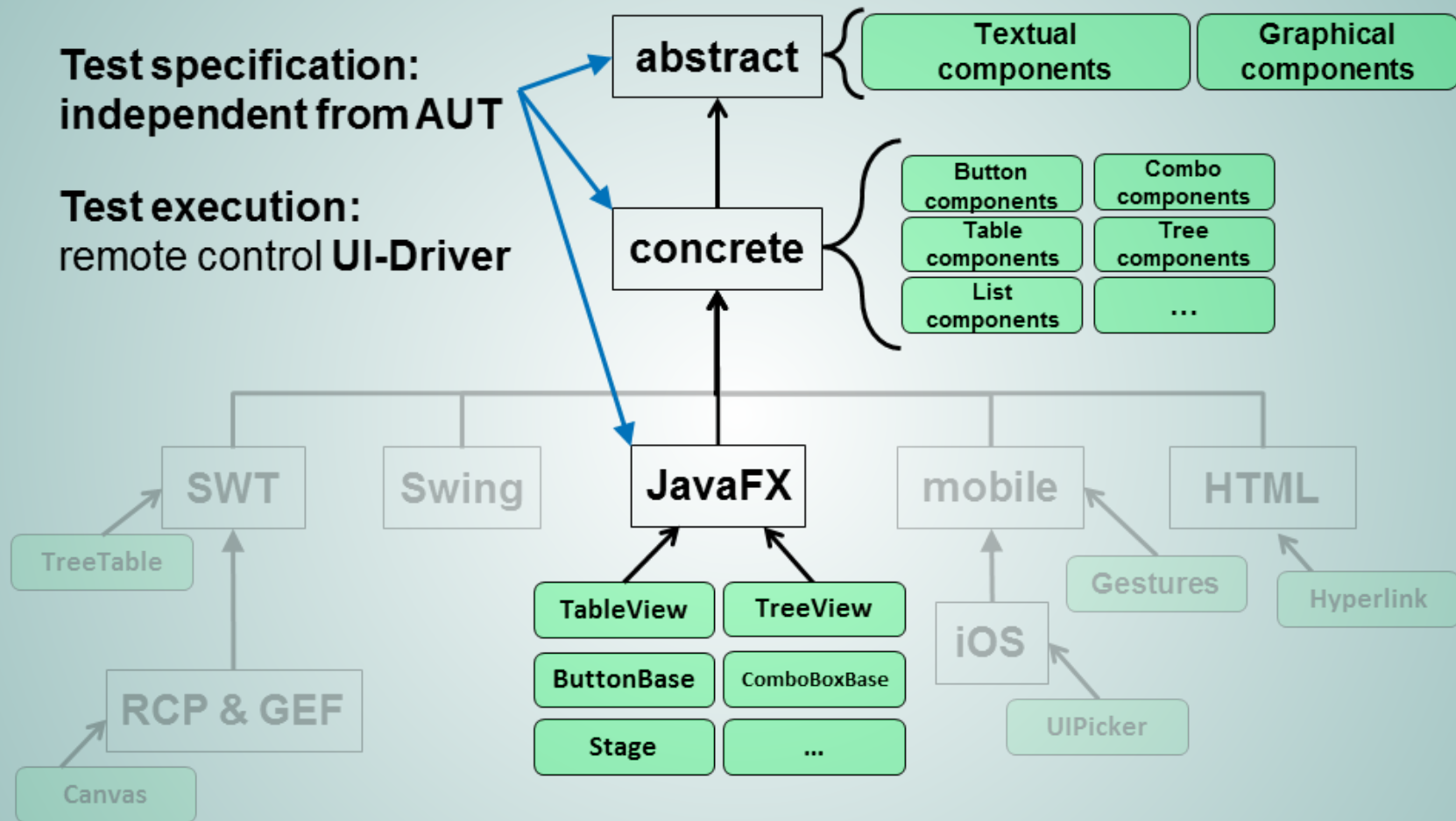
THE CONCEPT: TOOLKIT ABSTRACTION



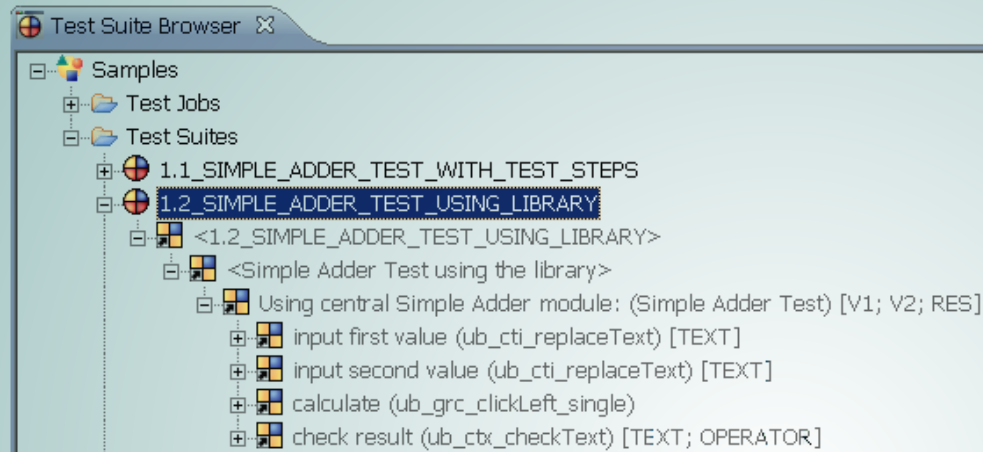
THE CONCEPT: TOOLKIT ABSTRACTION



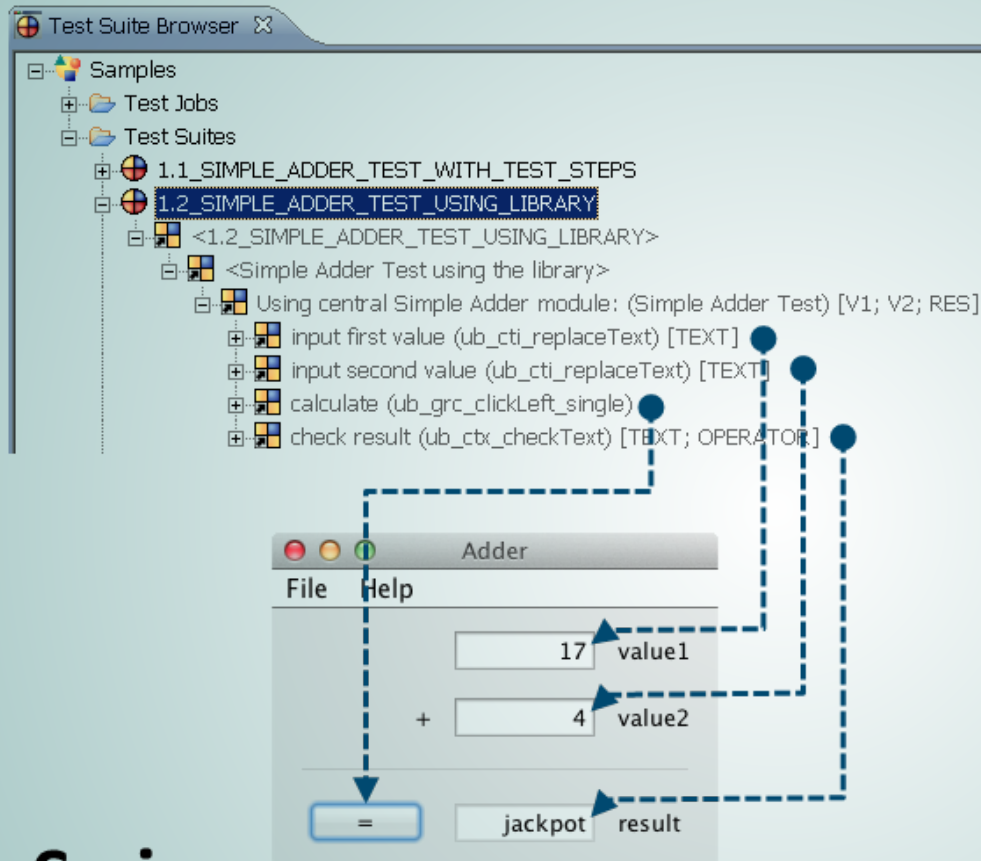
THE CONCEPT: TOOLKIT ABSTRACTION



THE CONCEPT: OBJECT MAPPING

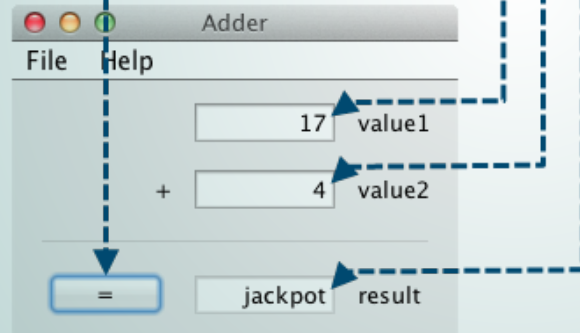
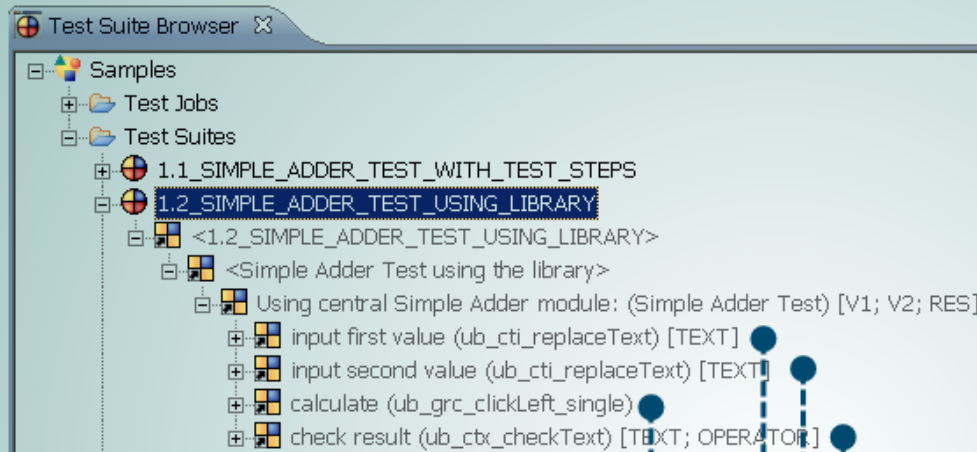


THE CONCEPT: OBJECT MAPPING

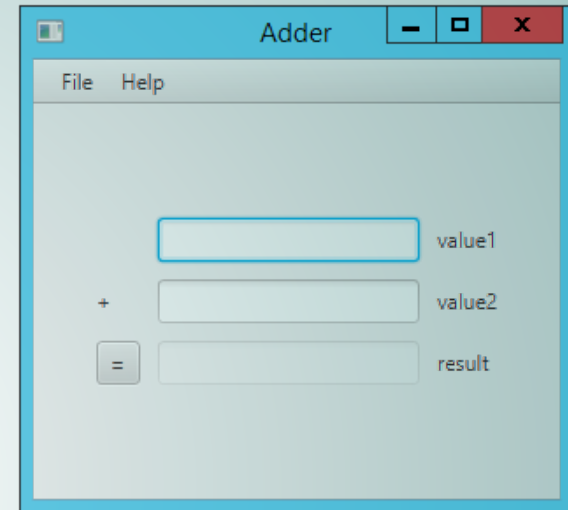


Swing

THE CONCEPT: OBJECT MAPPING

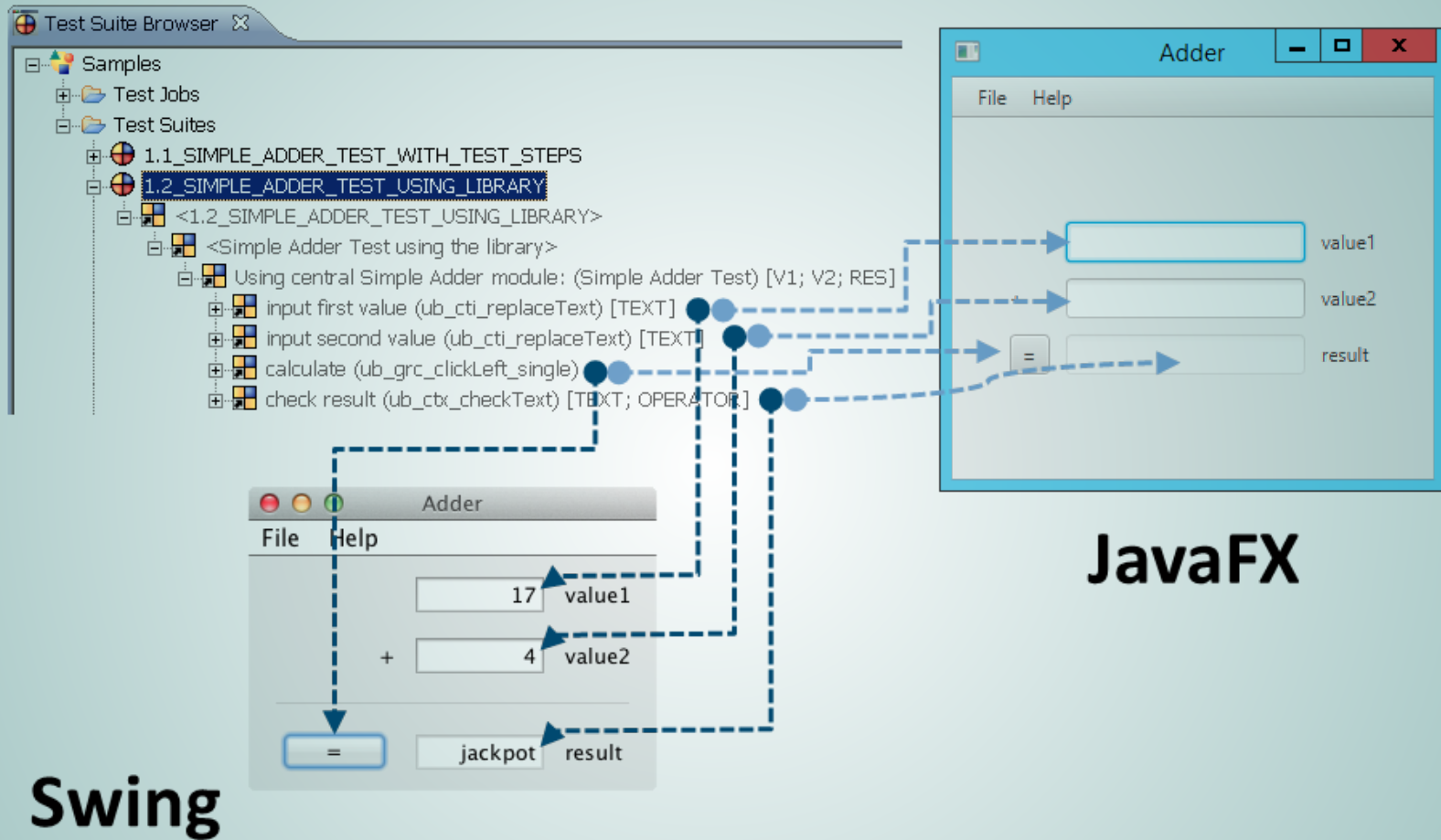


Swing

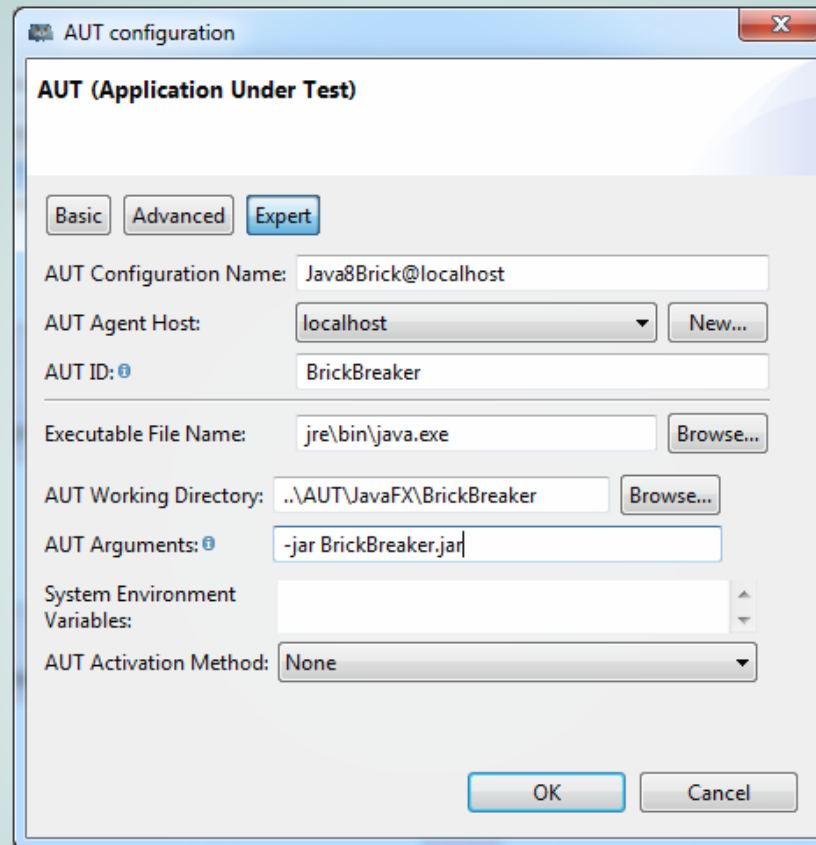


JavaFX

THE CONCEPT: OBJECT MAPPING



THE CONCEPT: AUT START



The image shows a screenshot of the 'AUT configuration' dialog box. The title bar reads 'AUT configuration' with a close button. The main title is 'AUT (Application Under Test)'. There are three tabs: 'Basic', 'Advanced', and 'Expert', with 'Expert' being the active tab. The configuration fields are as follows:

- AUT Configuration Name: Java8Brick@localhost
- AUT Agent Host: localhost (dropdown menu) with a 'New...' button.
- AUT ID: BrickBreaker
- Executable File Name: jre\bin\java.exe with a 'Browse...' button.
- AUT Working Directory: ..\AUT\JavaFX\BrickBreaker with a 'Browse...' button.
- AUT Arguments: -jar BrickBreaker.jar
- System Environment Variables: (empty text area)
- AUT Activation Method: None (dropdown menu)

At the bottom right, there are 'OK' and 'Cancel' buttons.

DEMO: SIMPLE ADDER

WHAT'S WORKING?

JavaFX on Java 8

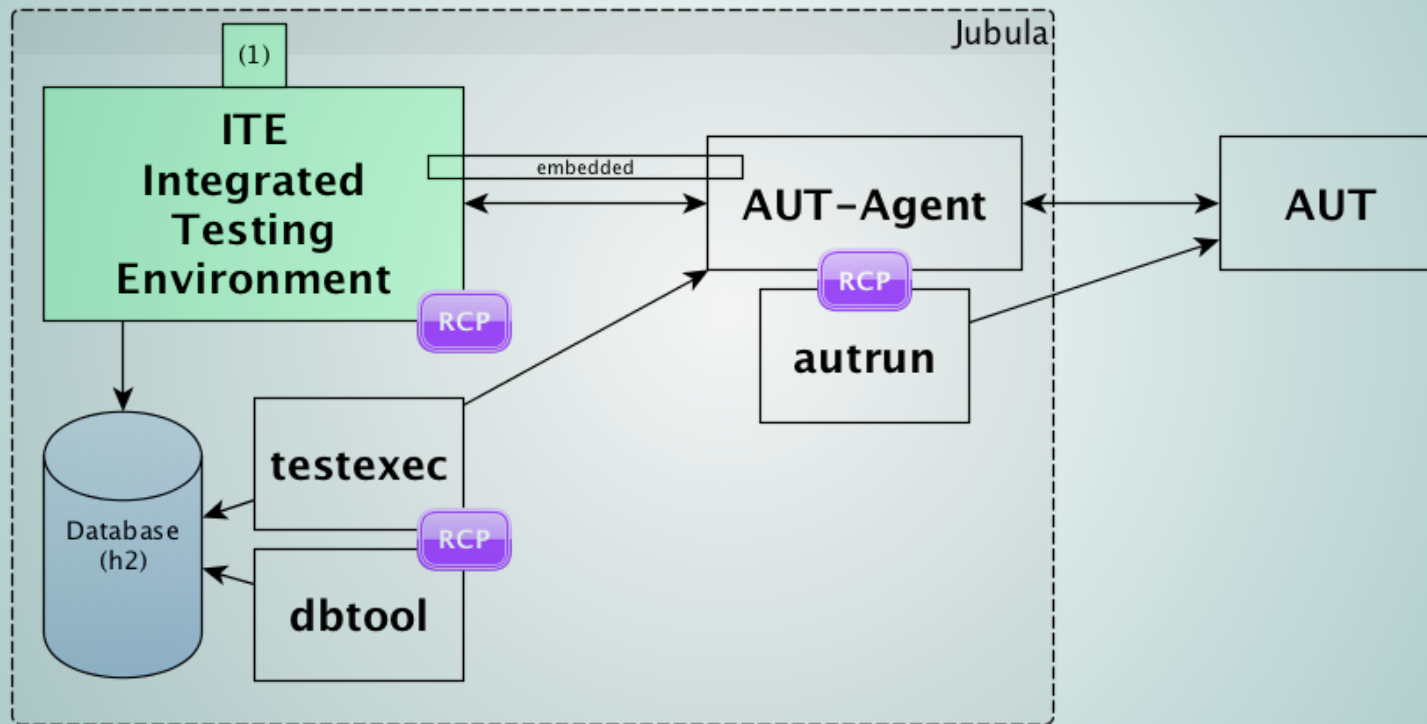
Standalone and **e(fx)clipse** based AUTs

Supported widgets: Accordion, Button, ButtonBase, CheckBox, ChoiceBox, ComboBox, Hyperlink, ImageView, Label, ListView, MenuButton, PasswordField, RadioButton, Stage, TableView, TabPane, Text, TextInputControl, TitledPane, TextArea, TextField, TreeView, TreeTableView, ToggleButton

current status: <http://www.eclip.se/421595>

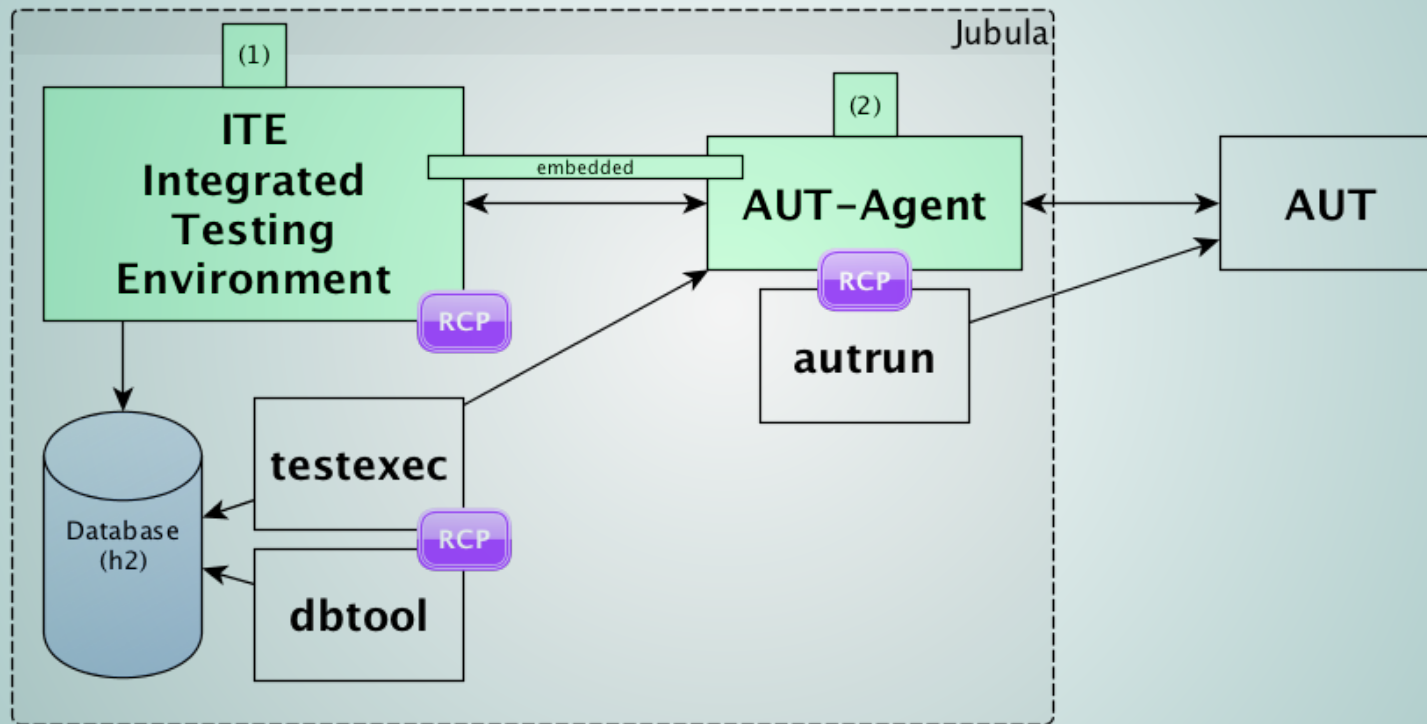
SUPPORTING CUSTOM COMPONENTS

(1) write extension for ITE: `o.e.jubula.toolkit.common.toolkitsupport`
information about type-system and supported CAPs



SUPPORTING CUSTOM COMPONENTS

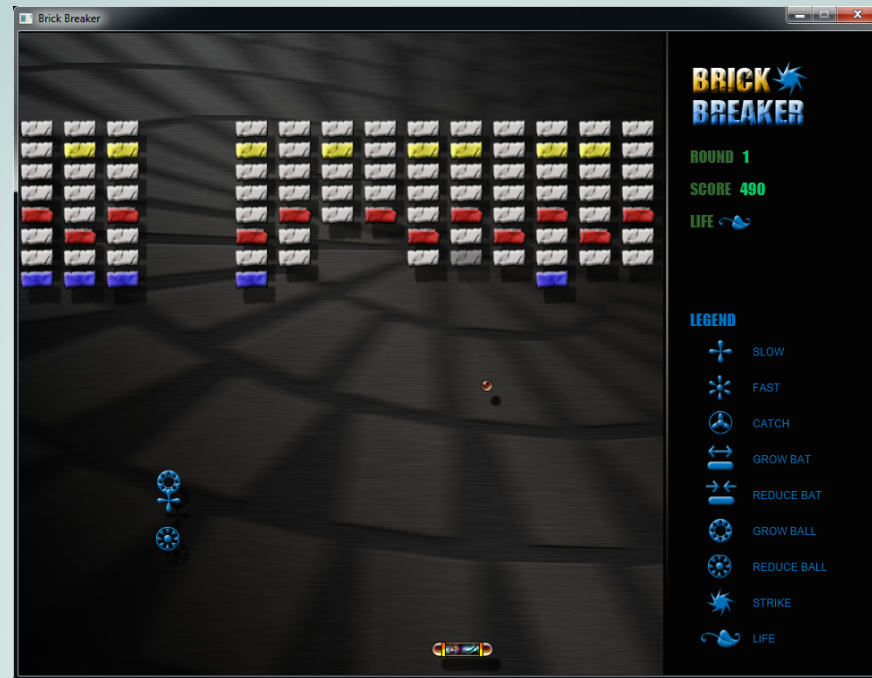
- (2) provide fragments for RC bundles in AUT-Agent
- provide toolkit dependent technical realization



LET'S HAVE SOME FUN...

- show 3 high-level tests using only 4 test steps (out of 270+)
 - store
 - wait
 - compare
 - move
- demonstrate strength in object recognition
 - reliability
 - speed

... AND BREAK STH!



official Oracle JavaFX example application

SOME INSIGHTS...

no modifications in AUT necessary
but to improve object heursitic in JavaFX use:

```
/**  
 * The id of this {@code Node}. This simple string  
 * identifier is useful for finding a specific Node  
 * within the scene graph ...  
 */  
void javafx.scene.Node.setId(String value)
```

ADDITIONAL INFO / GETTING STARTED?

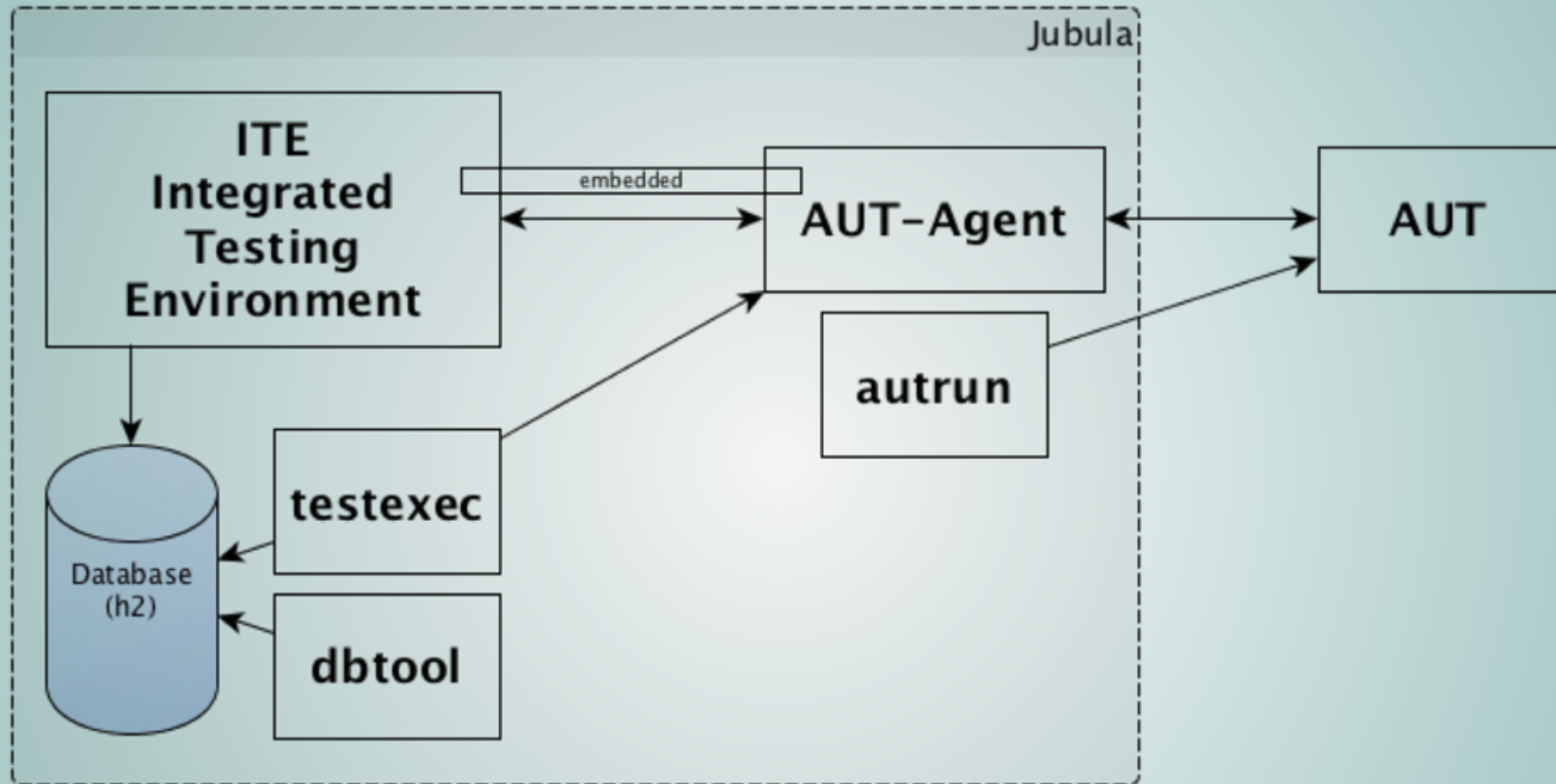
Get the latest version of Jubula!

Recommended: Eclipse for Testers Mars M3 - Friday, November 14

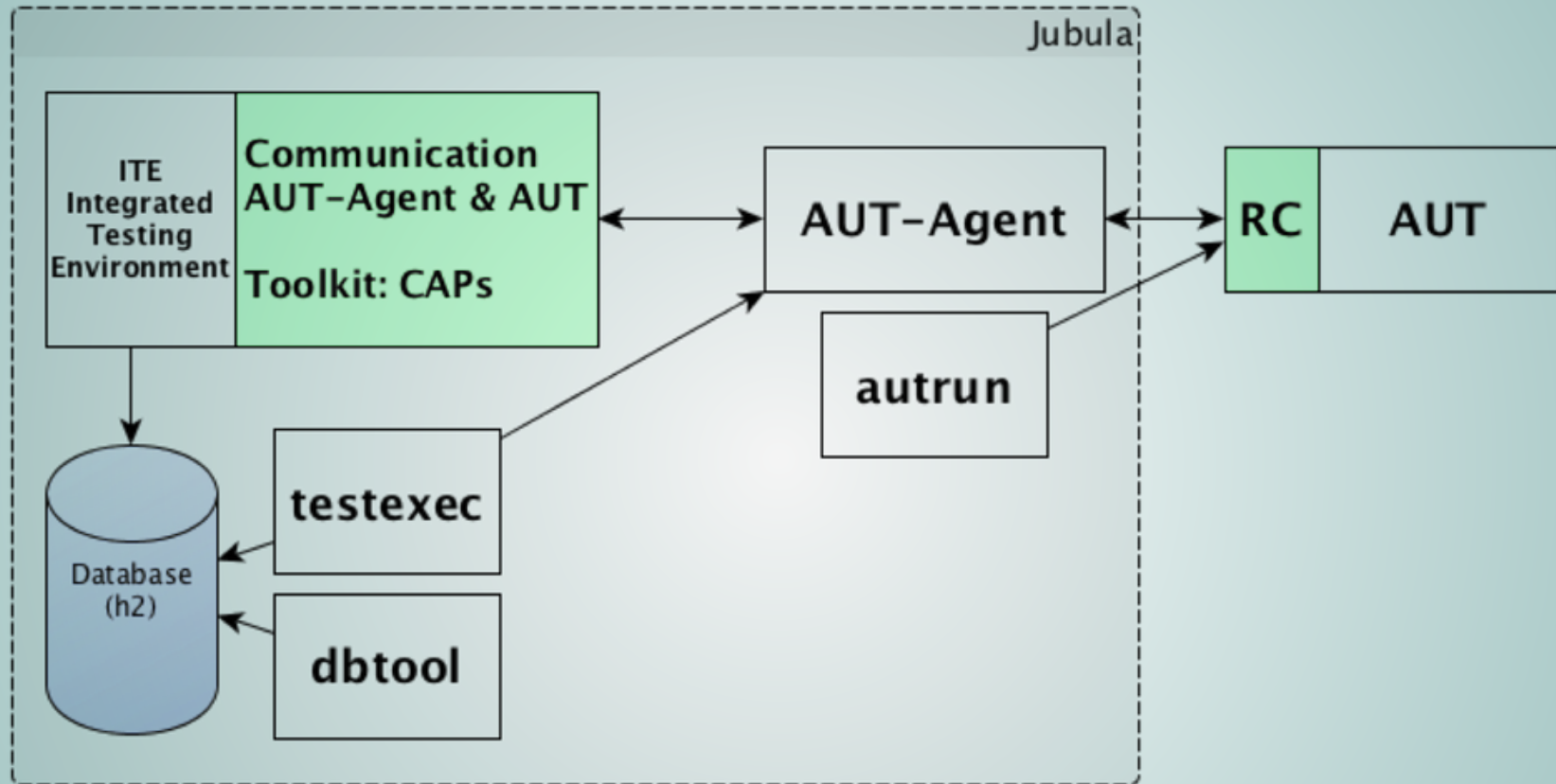
Eclipse Magazin - 5.2014 - Jubula goes JavaFX

First steps: Help → Cheat Sheets...

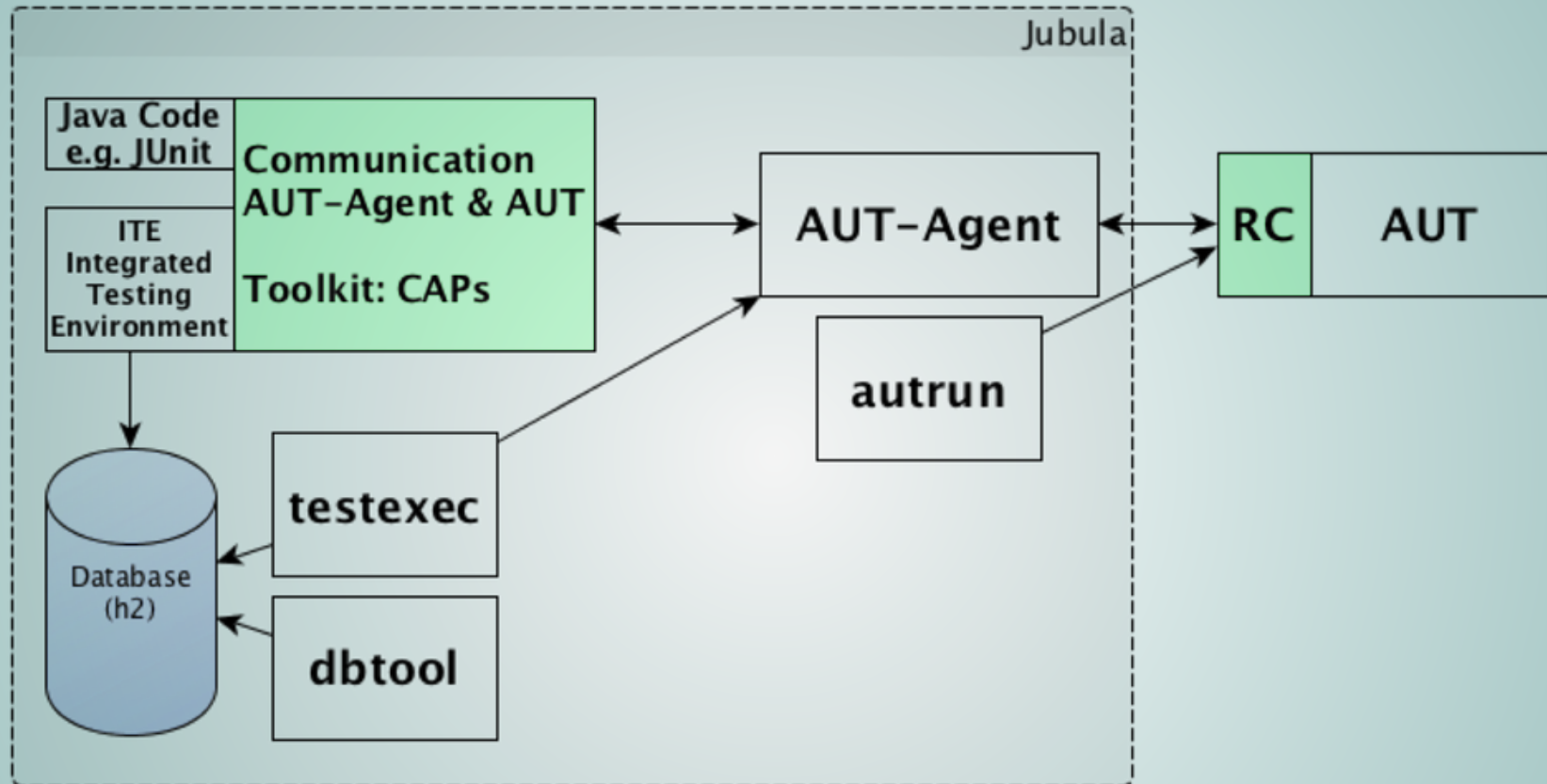
A SNEAK PEEK...



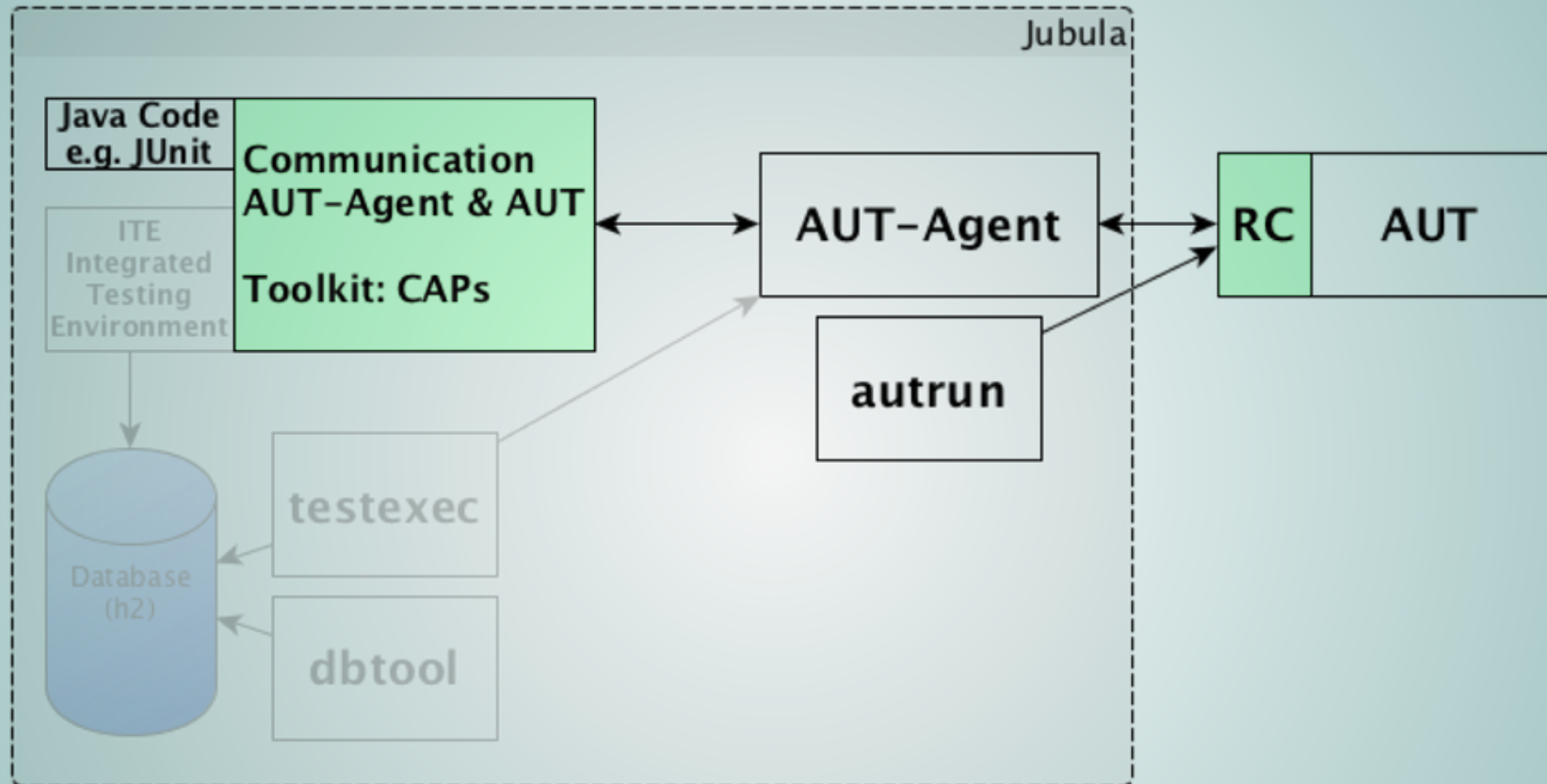
A SNEAK PEEK... TO AN API



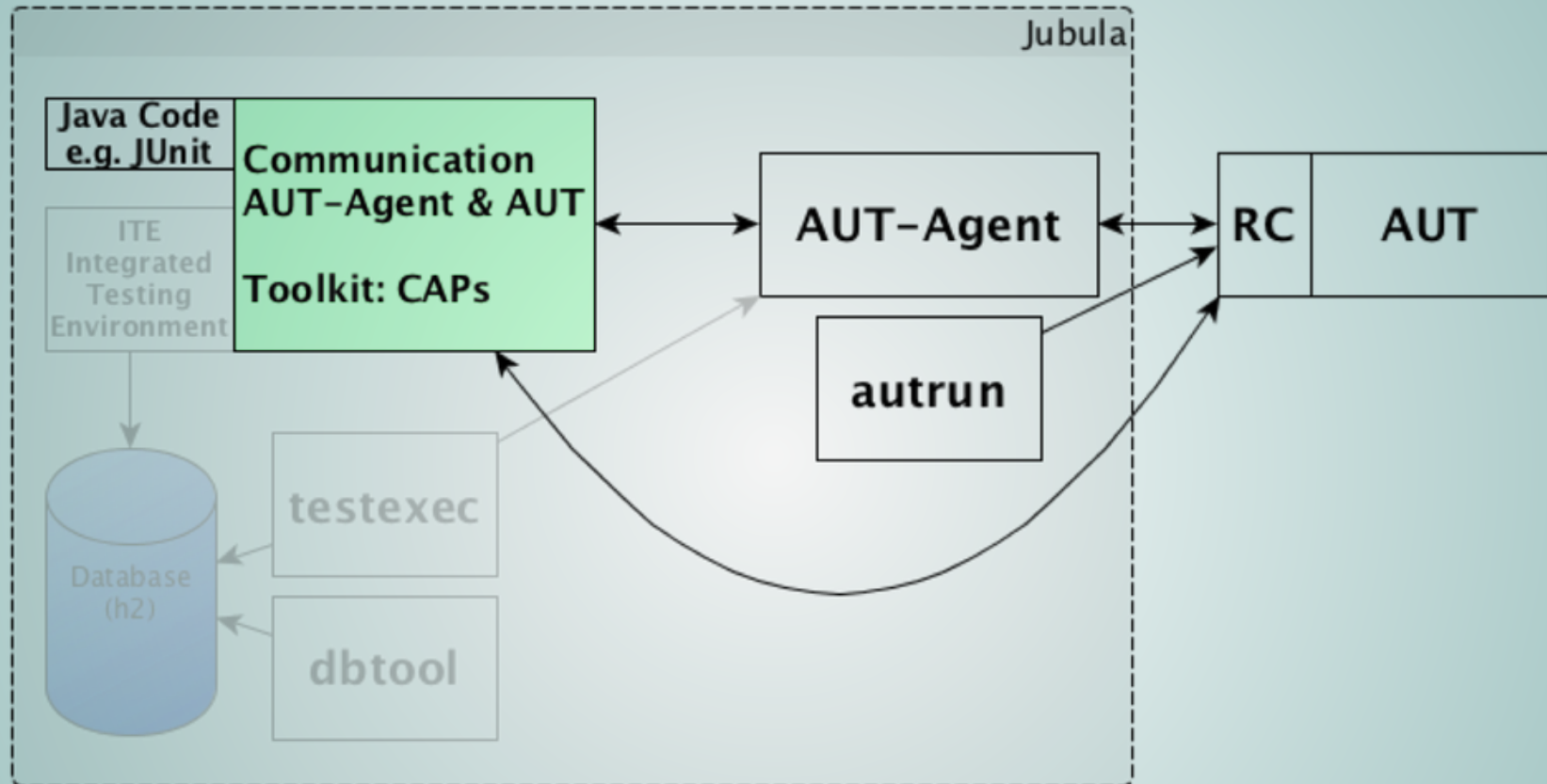
A SNEAK PEEK... TO AN API



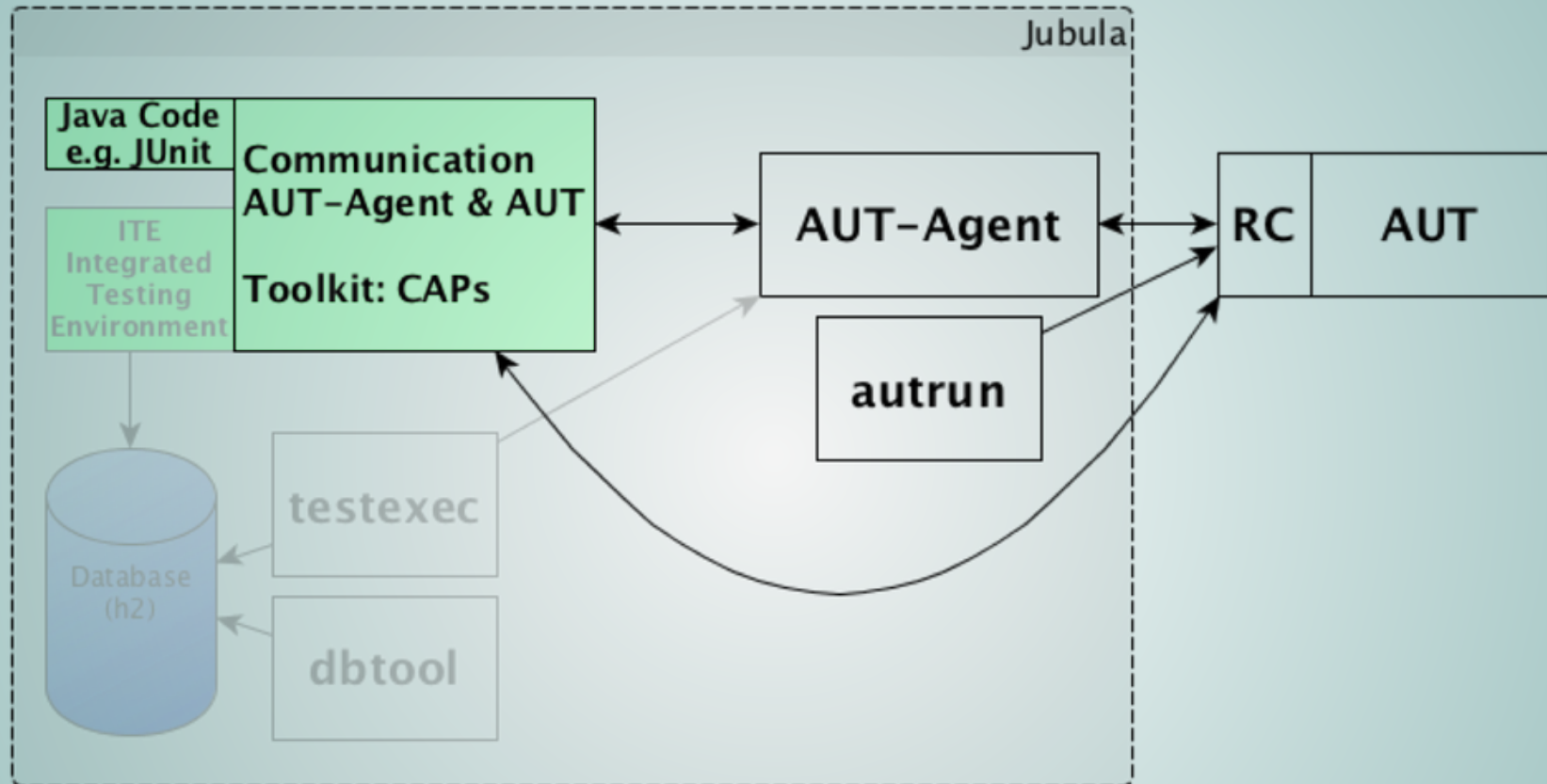
A SNEAK PEEK... TO AN API



A SNEAK PEEK... TO AN API



A SNEAK PEEK... TO AN API



Q&A...!?

Twitter: @MarkusTiede
Markus.Tiede@bredex.de

Copyright BREDEX GmbH 2014. Made available under the Eclipse Public License v1.0.