

Jekyll and Hyde with Jubula

Eclipse Demo Camp Munich

Alex Schladebeck

BREDEX GmbH



@alex_schl
alexandra.schladebeck@bredex.de

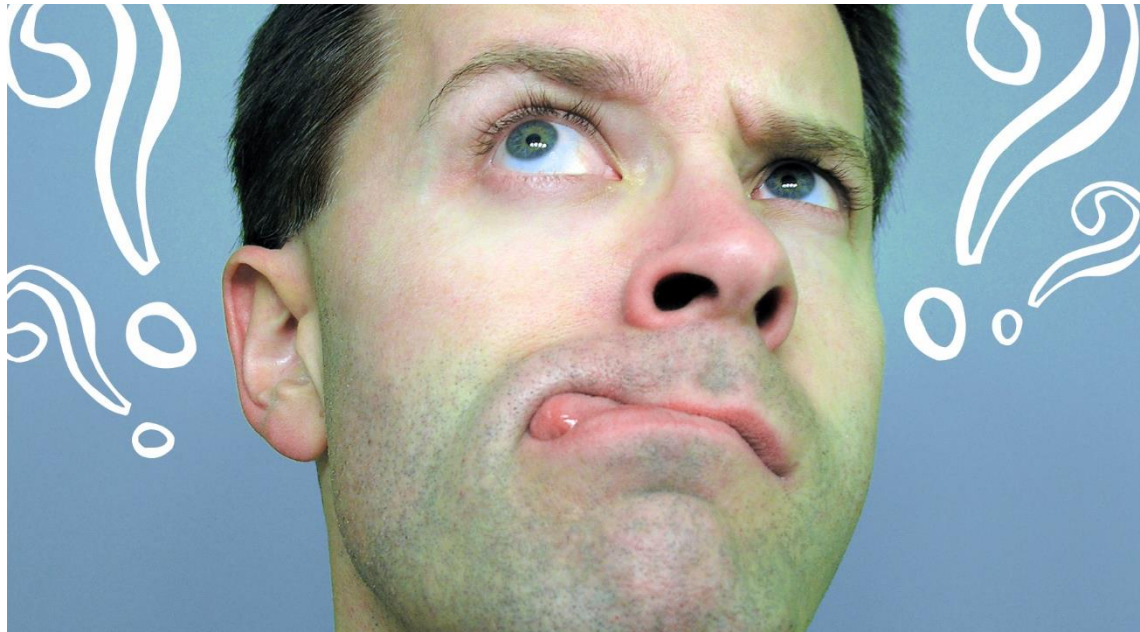


20 minutes of test automation!

- ▶ Intro to Jubula
- ▶ Dr. Jekyll: The Jubula ITE
- ▶ Mr. Hyde: The Jubula Client API

After-break wake-up call

► Why do we test?



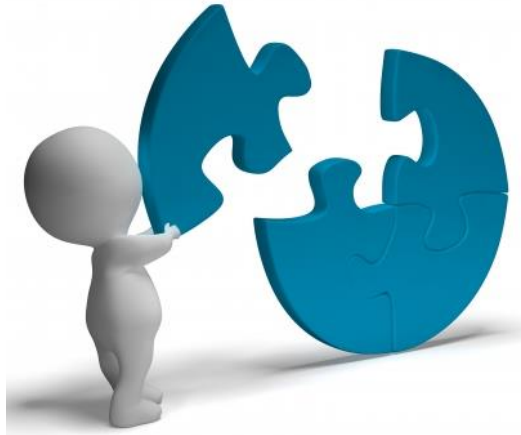
► Why do we automate tests?



Automating tests with Jubula

- ▶ Code-free functional (black box) testing
 - Swing, SWT/RCP/GEF, HTML, iOS, JavaFX
- ▶ Tests *through* the GUI
 - No / minimal difference to manual test
 - All layers tested
 - Use cases, workflows: begins with requirements
 - Intelligent and well-structured tests
- ▶ Communication!
 - Whole team involvement
 - Continuous Integration = Continuous Information
- ▶ “Why can’t we also write code...?”

CAP: The quintessence of automation



CAP (Test Step)

Component
(Which)

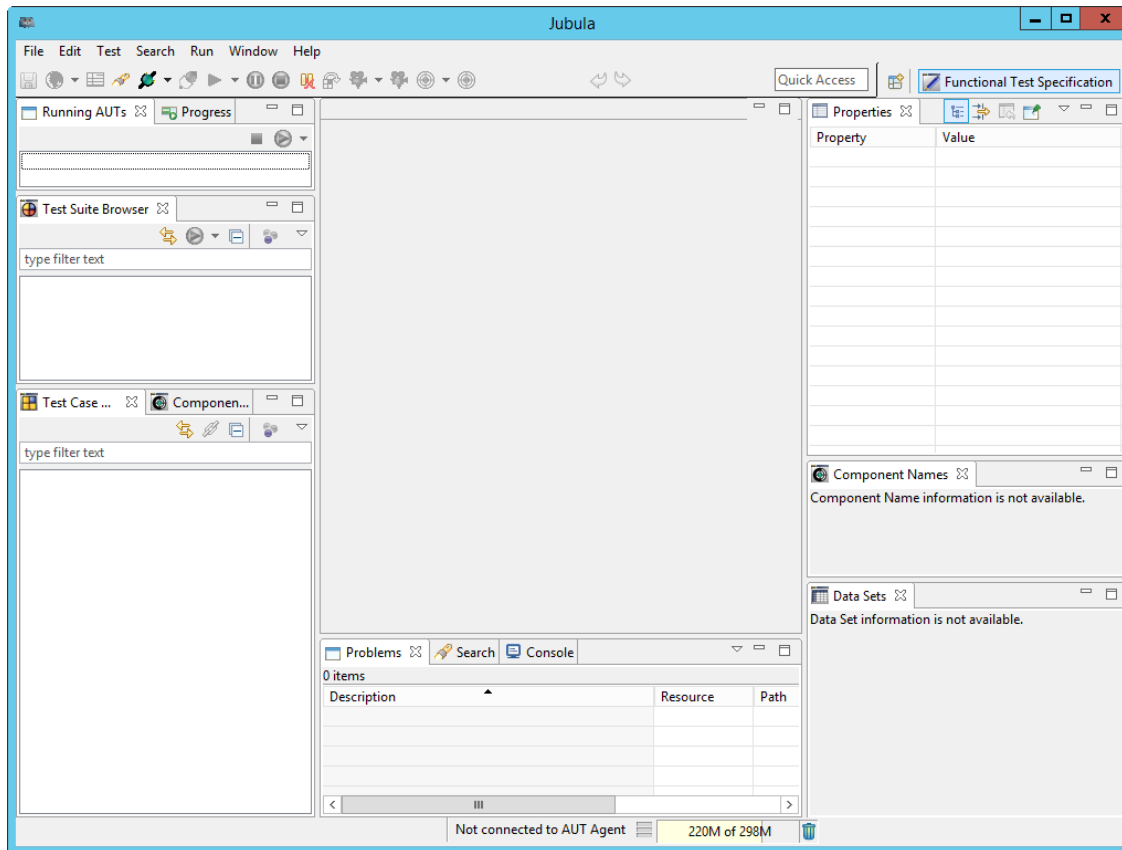
Action
(What)

Parameter
(How)

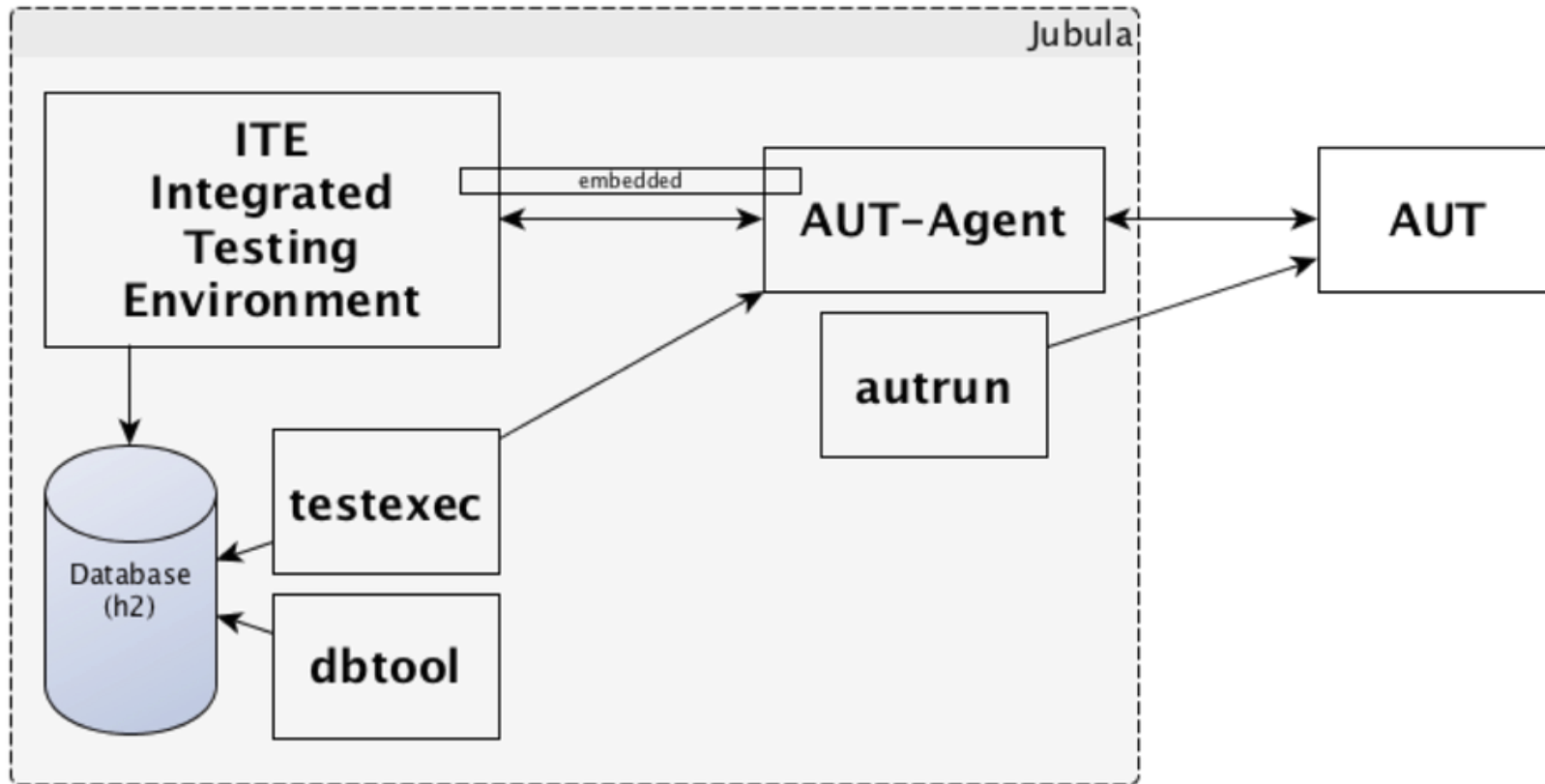
Username

Into the **username field**, enter **"admin"**

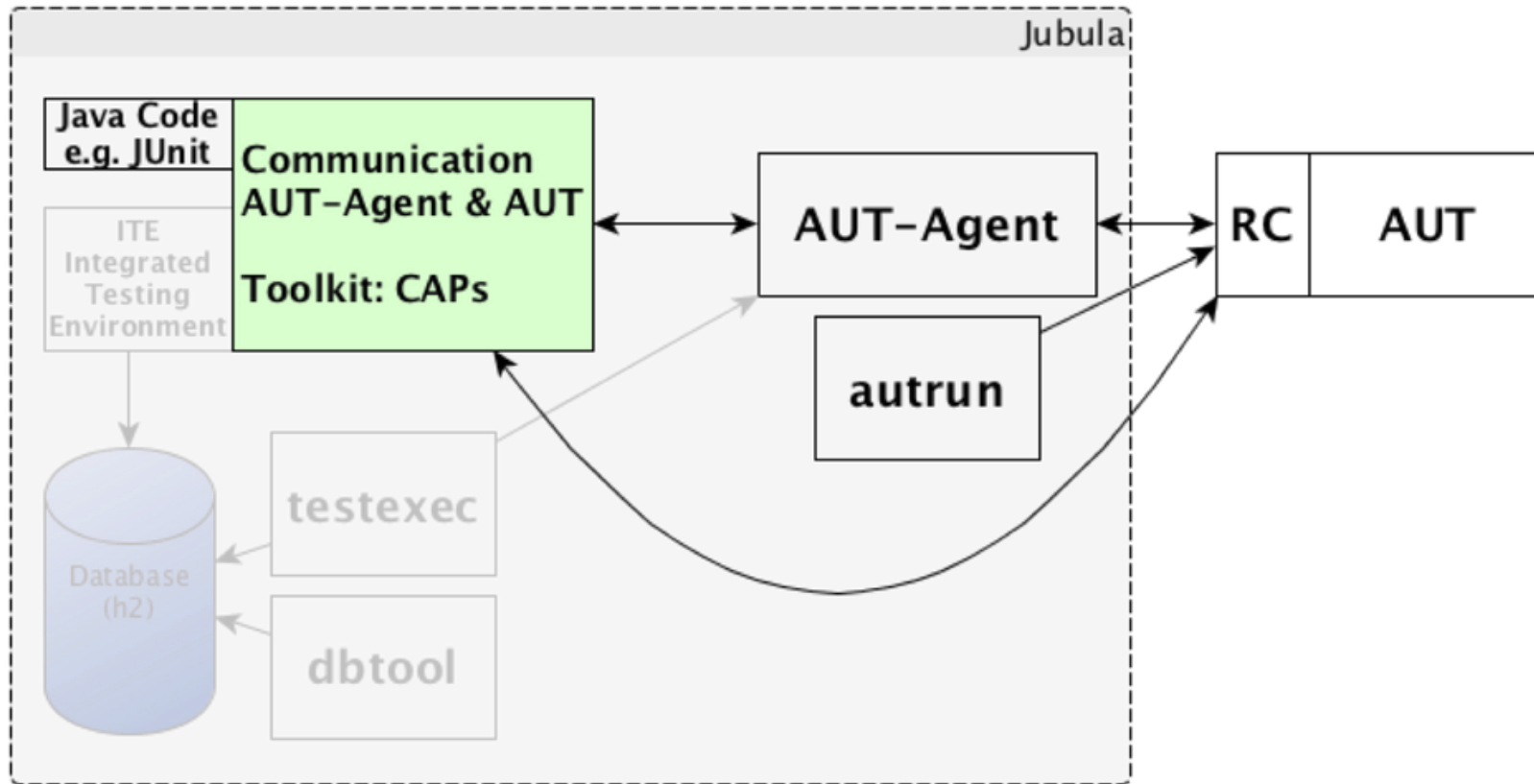
Demo ITE (Dr. Jekyll)



From ITE to IDE



From ITE to IDE



Jubula Client API (Mr. Hyde)

```
/** prepare */
@Before
public void setUp() throws Exception {
    m_agent = MakeR.createAUTAgent(AGENT_HOST, AGENT_PORT);
    m_agent.connect();

    final String autID = "SimpleAdder_rcp"; //$NON-NLS-1$
    AUTConfiguration config = new RCPAUTConfiguration(
        "api.aut.conf.simple.adder.rcp", //$NON-NLS-1$
        autID,
        "SimpleAdder.exe", //$NON-NLS-1$
        "..\\examples\\AUTs\\SimpleAdder\\rcp\\win32\\win32\\x86\\", //$NON-NLS-1$
        null,
        Locale.getDefault(),
        Locale.getDefault());

    AUTIdentifier id = m_agent.startAUT(config);
    if (id != null) {
        m_aut = m_agent.getAUT(id, SwtComponents
            .getToolkitInformation());
        m_aut.connect();
    } else {
        Assert.fail("AUT start has failed!"); //$NON-NLS-1$
    }
}
```



Using the API

- ▶ Java 6+
- ▶ Same actions and concepts as in the ITE
- ▶ Exceptions for execution events
- ▶ Consumable in different flavours
 - Plain JARs
 - Maven artifacts from m2-repo
 - OSGi bundles
- ▶ Small footprint ~2MB
- ▶ In Jubula standalone and coming with Mars
 - Standalone version <http://testing.bredex.de>

Dual personalities

▶ Jubula ITE

- Code free
 - + test design
- Communication
- Shiny extras
 - ALM
 - Test result reports
 - BIRT

▶ Jubula Client API

- Very code-y
 - + coding practices ;)
 - With great power comes great responsibility...
- Communication
- Shiny features
 - Complex test flows
 - Use the world of Java
 - Integration into SCM



Choose your poison!