

Saros Test Framework für die HTML GUI

Antrittsvortrag Masterarbeit

Jannis Fey

24.07.18

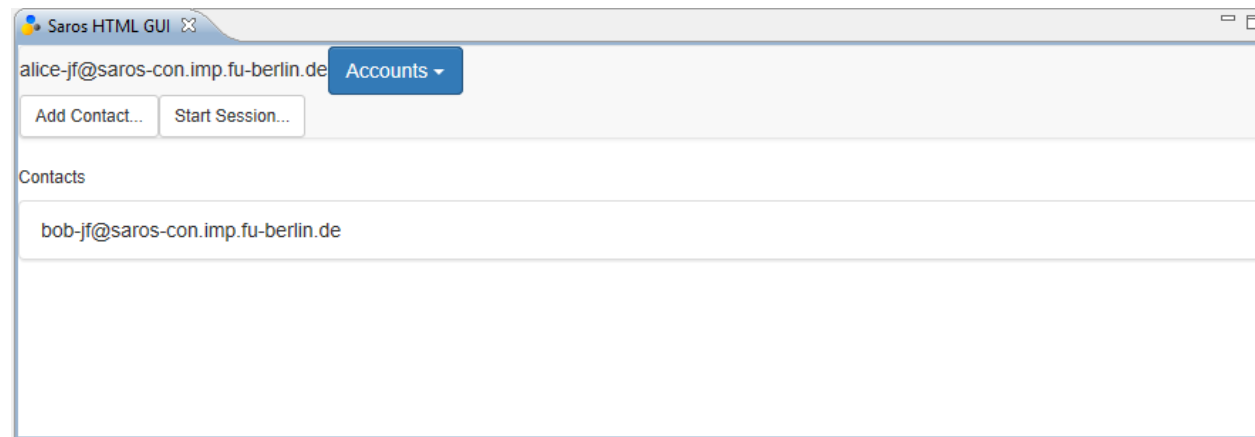
Was ist Saros?

- Plugin für Eclipse & IntelliJ
- Ermöglicht verteilte Paarprogrammierung in Echtzeit



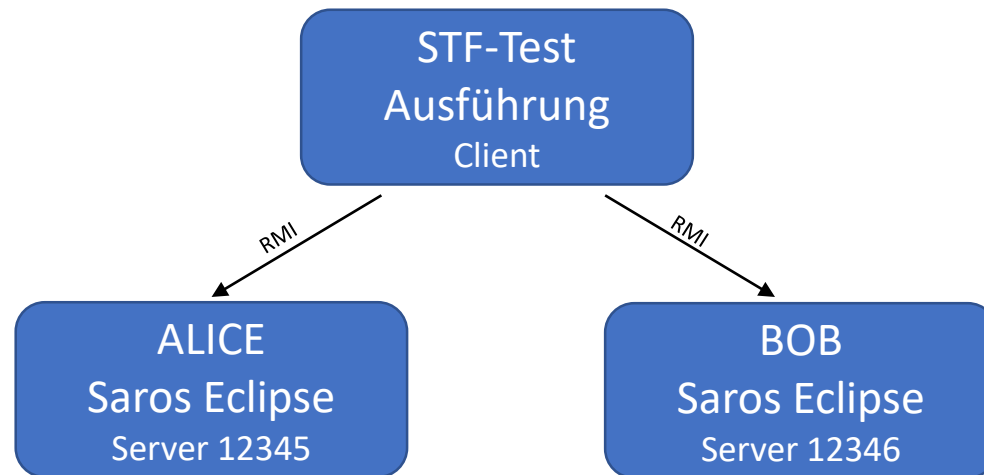
HTML-GUI

- IDE unabhängig
- In React geschrieben
- Wird in einem eingebetteten Browser angezeigt



Saros Test Framework (STF)

- Eigenes Framework um Saros zu testen
- Ermöglicht Ende-zu-Ende test
- Interagiert direkt mit der GUI
- Interaktion mit HTML-GUI war bisher nicht möglich



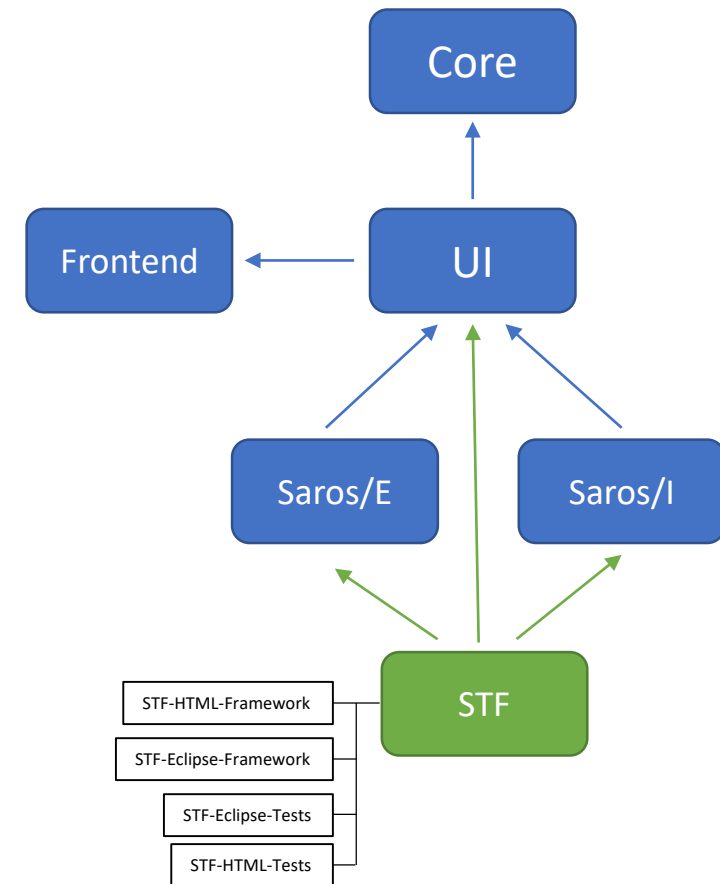
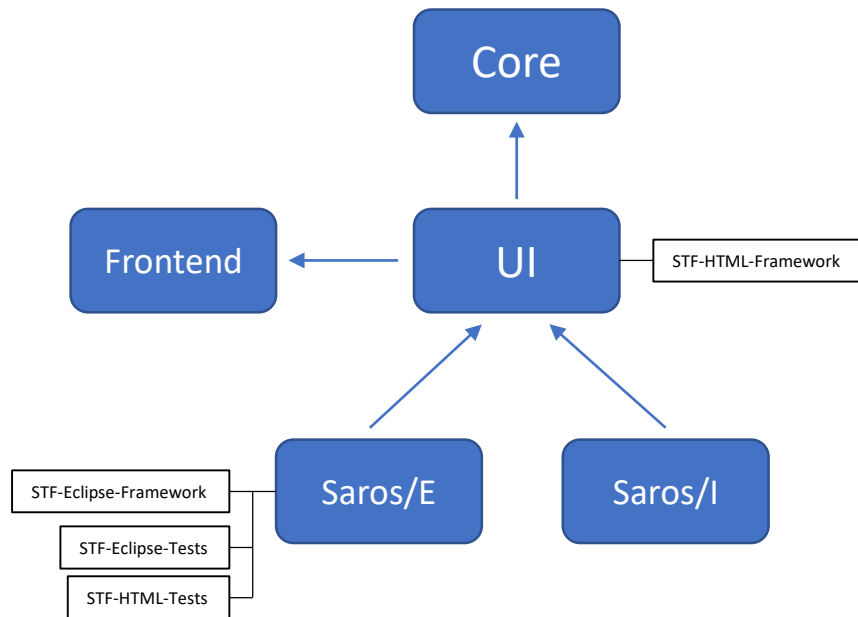
Zielsetzung

- Interaktion mit HTML-Elementen
 - Button, Checkbox, Input, ...
- Modularisierung
 - STF als eigenständige Komponente
- Bestehende STF-Test für HTML-GUI anpassen
- STF-Tests mit IntelliJ starten

Interaktion mit HTML-Elementen

- Via Javascript Aufruf kann interagiert werden
- JQuery ist eingebunden
- Beispiele:
 - `browser.run("$('#add-contact-btn').click()');`
 - `browser.run("$('input[name=jid]').value('alice@fu-berlin.de')');`
 - `browser.syncRun("return view.getFieldValue('checkbox')');`

Modularisierung



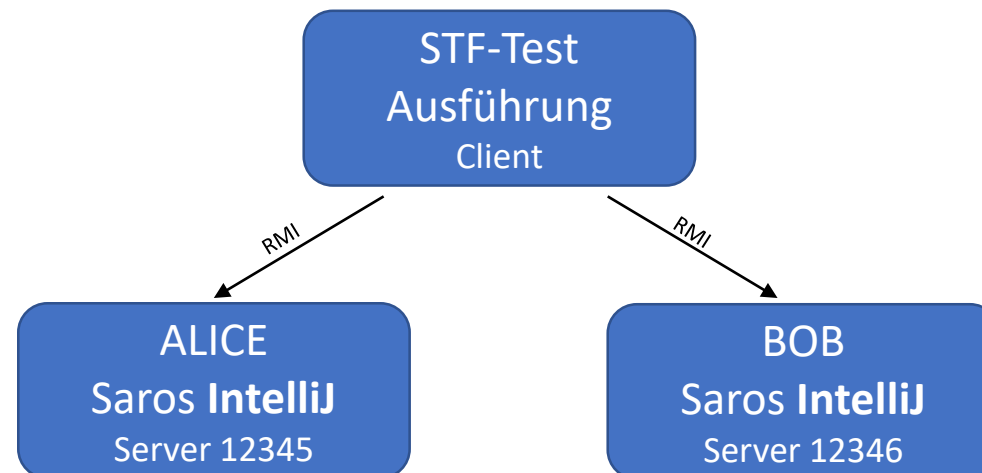
Bestehende STF-Test für HTML-GUI anpassen

- Bestehenden Test kopieren
- Kopie für HTML-GUI ausführbar machen

- Abgrenzung
 - Nur Infrastruktur
 - Es ist okay wenn erstmal „alle“ kopierten Test fehlschlagen, dafür aber die HTML-GUI benutzen

STF-Tests mit IntelliJ starten

- Ein purer HTML-STF Test sollte erfolgreich durchlaufen
- Infrastruktur schaffen um Test für Eclipse oder IntelliJ zu starten
- Abgrenzung
 - Swing Elemente von IntelliJ (Menü, etc.) nicht ansprechen



Meine Fragezeichen

- Wie genau soll die Infrastruktur für kopierten Test aussehen
 - Eigene Annotation `@HtmlTest`
 - Namenkonvention „xyzTest_html“
 - Initialisierung `ALICE.useHtmlGUIForThisTest()`
- Wie umgehen mit Lücken in HTML-GUI
 - z.B. STF-Test für den Chat

Live Demo Backup

Text

Email

Password

Checkbox 1 2 3

Radio 1 2 3

Select

MultiSelect

ProgressBar

Pressed Button: key-split-1

```
@Test
public void shouldTestCheckboxes() throws Exception {
    IRemoteHTMLView view = ALICE.htmlBot().view(BASIC_WIDGET_TEST);

    assertTrue(view.hasElementWithName("checkbox1"));
    assertTrue(view.hasElementWithName("checkbox2"));
    assertTrue(view.hasElementWithName("checkbox3"));

    view.checkbox("checkbox1").check();
    view.checkbox("checkbox2").check();
    view.checkbox("checkbox2").unchecked();
    view.checkbox("checkbox3").check();

    assertTrue(view.checkbox("checkbox1").isChecked());
    assertFalse(view.checkbox("checkbox2").isChecked());
    assertTrue(view.checkbox("checkbox3").isChecked());
}
```