

# Bestandsaufnahme und Arbeit an einer Alpha-Version des Saros- Plugins für die IntelliJ-Plattform

Verteidigung der Bachelorarbeit

# Agenda

Saros-Plugin

Entwicklung von Saros

Ziel: Saros/I-Alpha-Version

Ergebnisse im Überblick

Saros/I-Ressourcenverwaltung

Evaluation

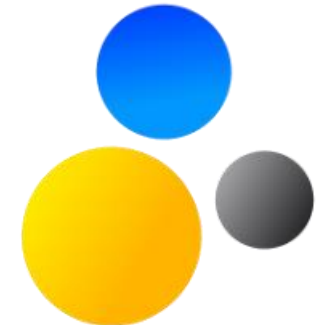
Ausblick

# Saros-Plugin

- Tool um verteilte Paar-Programmierung zu ermöglichen
- Mögliche Vorteile von Paar-Programmierung:
  - Nur geringer Mehraufwand
  - Kann Codequalität erhöhen
  - Kann Wissensaustausch zwischen den Entwicklern erlauben
- Verteilte Paar-Programmierung – Ersetzen der direkten Kommunikation durch Hilfsmittel
  - Hilfsmittel wie geteilte Dokumente, Text-, Sprach- oder Videokommunikation
  - Kann selben Vorteile wie normale Paar-Programmierung erzielen
- Saros erlaubt das parallele Arbeiten mehrerer Personen an geteilten Dateien

# Entwicklung von Saros

- Projekt an der Freien Universität Berlin
- Beginn 2006 über Diplomarbeit „Entwicklung einer Eclipse-Erweiterung zur Realisierung und Protokollierung verteilter Paarprogrammierung“
- Ausweitung auf andere Entwicklungsumgebungen
  - Aufteilung von bestehendem Saros-Plugin in **Saros Core** und **Saros/E**
- Entwicklung von **Saros/I** für IntelliJ IDEA (und **Saros/N** für Netbeans)
- Code-Qualität der IntelliJ-Version unzureichend
  - Code-Duplikate
  - Defekte
  - Fehlende Dokumentation und Testabdeckung



# Ziel: Saros/I-Alpha-Version

- Erfassen der Anforderungen an eine Saros-Alpha-Version
- Erstellen und Veröffentlichen einer funktionierenden Alpha-Version von Saros/I

## Meilensteine:

1. Die grundlegende Funktionalität von Saros/I ist vorhanden.
2. Die grundlegende Funktionalität der Saros-HTML-GUI ist vorhanden und in Saros/I integriert.
3. Die erste Alpha-Version ist vollständig und veröffentlicht.

# Ergebnisse im Überblick

- Dokumentation von bestehenden Versagen in Saros/I
- Behebung von Defekten in Saros/I:
  - Module korrekt teilen
  - Nach dem Start einer Sitzung synchron auf geteilten Dateien arbeiten
  - Auf geteilten Dateien arbeiten, welche
    - Nicht bei allen Teilnehmern geöffnet sind
    - Als Teil der Sitzungs-Erstellung übertragen wurden
  - Während einer Sitzung nicht-geteilte Dateien öffnen
- Reimplementierung der Saros/I-Ressourcenverwaltung

# Saros-Ressourcenverwaltung Überblick

- Umgang mit geteilten Projekten (Modulen), Dateien und Ordnern
- Umsetzung von Schnittstellen des Saros-Kerns
  - Schnittstellen stark an ursprünglicher Eclipse-Implementierung orientiert

# Saros/I-Ressourcenverwaltung

## Designunterschiede Eclipse – IntelliJ

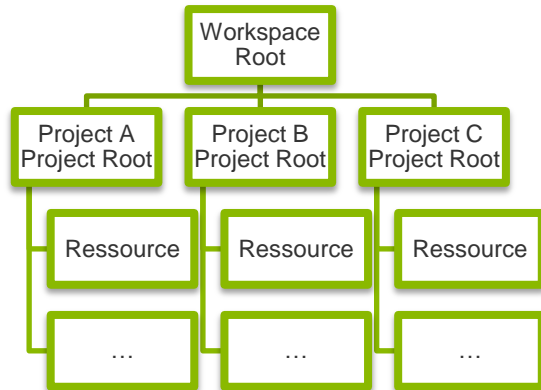
- Verweise auf Ressourcen
  - Eclipse verwendet virtuelle Verweise auf Ressourcen
  - IntelliJ kann nur Verweise auf vorhandene Ressourcen erstellen
- Saros/I versucht für jede zu behandelnde Ressource einen Verweis zu erstellen
  - Über IntelliJ-API nicht möglich
  - ➔ IntelliJ-API wird zur Ressourcenverwaltung in Saros/I nicht verwendet
    - ➔ Erhöht Implementierungsaufwand und Fehleranfälligkeit



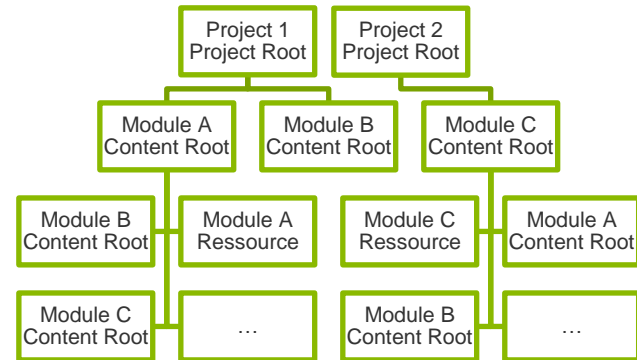
# Saros/I-Ressourcenverwaltung Designunterschiede Eclipse – IntelliJ

- Pfade von Projekten
  - In Eclipse hat jedes Projekt ein Grundverzeichnis (Project Root)
  - In IntelliJ hat jedes Modul ( $\equiv$  Eclipse Projekt) beliebig viele Grundverzeichnisse

## Eclipse



## IntelliJ

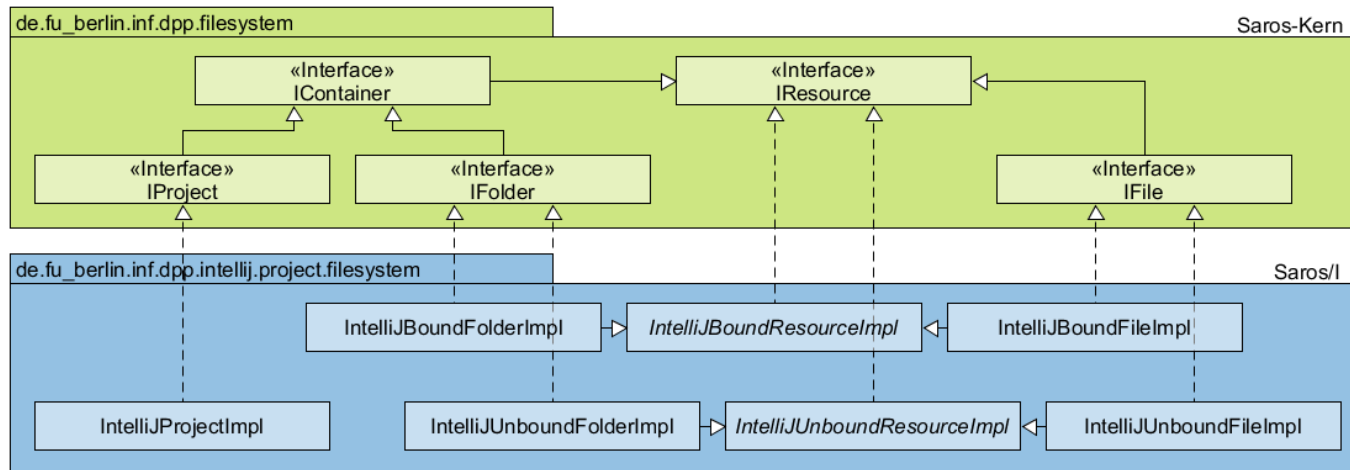


- Saros/I-Ressourcenverwaltung basiert auf Eclipse-Modell
  - IntelliJ-API wird nicht verwendet  $\rightarrow$  Content Root der Module immer direkt unterhalb der Project Root angenommen

# Saros/I-Ressourcenverwaltung

## Design der Reimplementierung

- Implementierung als Adapter für die IntelliJ-API
- Orientierung an der vom Saros-Kern erwarteten Funktionalität
- Spaltung der Ressourcenverwaltung
  - Gebundene Implementierung (wirklicher Verweis)
  - Ungebundene Implementierung (virtueller Verweis)



# Evaluation

- Schwerpunkt hat sich im Verlauf der Arbeit geändert
  - Reimplementierung der Ressourcenverwaltung zentrales Problem
- Fortschritt in Arbeit an einer Saros/I-Alpha-Version
- Schafft gute Grundlage für weitere Arbeit
  - Dokumentation der zu behebenden Versagen
  - Verbesserung der Code-Qualität und -Dokumentation

# Ausblick

- Nächste Schritte:
  - Integration der neuen Saros/I-Ressourcenverwaltung
  - Anpassung des Umgangs mit Änderungen der geteilten Dateistruktur an neue Ressourcenverwaltung
  - Behebung der erkannten Versagen
- Weiterentwicklung und Integration der Saros-HTML-GUI
  - Zur Zeit in Arbeit als Teil einer anderen Bachelorarbeit (Marius Schidlack)
- Dokumentation und Veröffentlichung der Saros/I-Alpha-Version
- Mögliche Erweiterungen – Ausbau des Saros/I-Funktionsumfangs
  - Aufheben der für die Alpha-Version angenommenen Einschränkungen
  - Erweitern um in Saros/E vorhandene Funktionalitäten

**Vielen Dank für Ihre Aufmerksamkeit!**

**Fragen?**