

Abschlussvortrag Masterarbeit

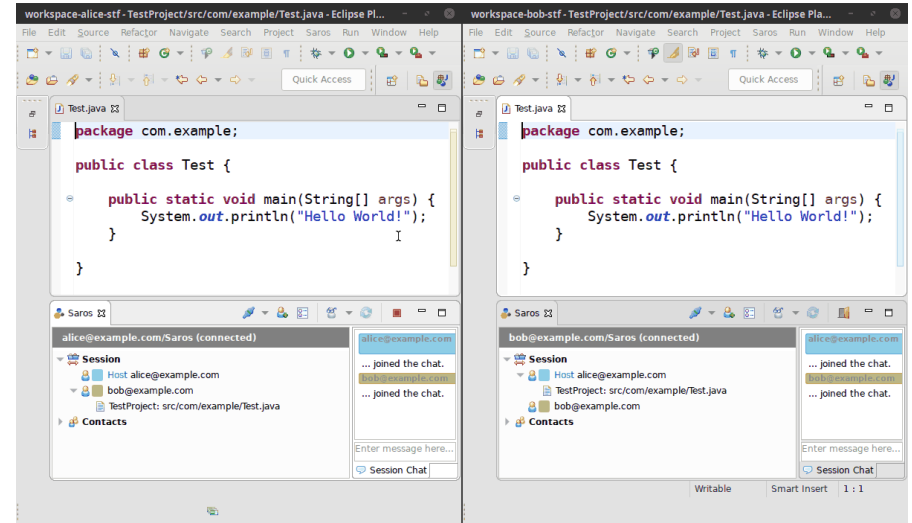
Thema: Refactoring des Saros-Filesystems

Gliederung

1. Problembeschreibung
2. Referenzpunktbasierte Architektur
3. Vorgehen/Strategie
4. Ergebnisse
5. Bewertung
6. Fazit

Was ist Saros?

- IDE Plugin für gemeinsame, verteilte Paarprogrammierung in Echtzeit
- Projekte können geteilt werden



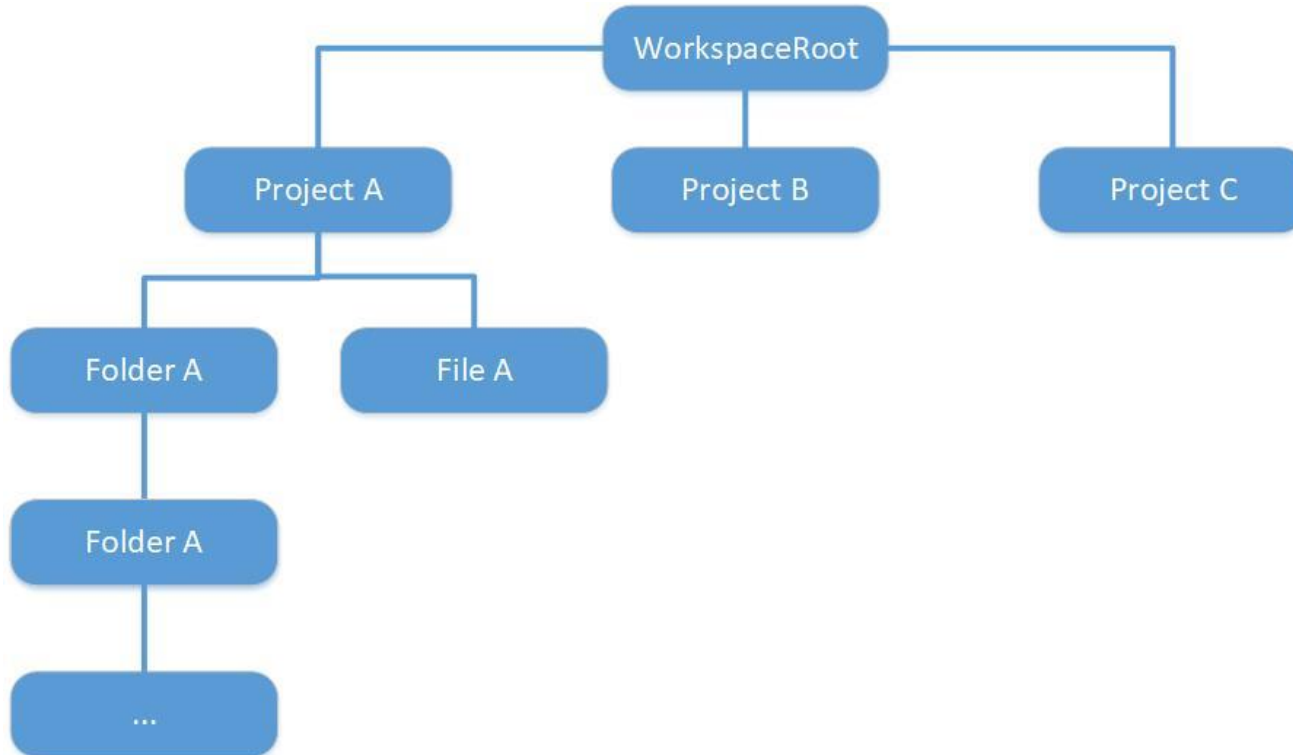
Quelle:

https://github.com/saros-project/saros/blob/master/docs/assets/images/animation/this_is_saros.gif

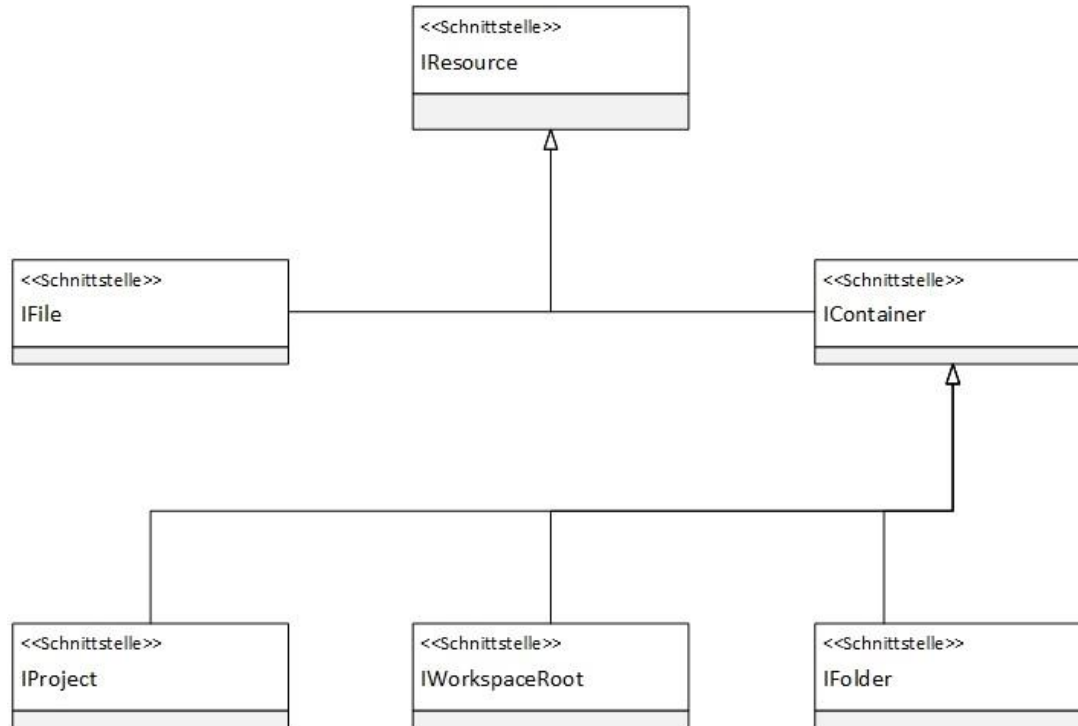
Werdegang des Problems

- Machbarkeitsstudie: Saros for IntelliJ (Saros/I)
 - Viele technische Schulden
- Unabhängige Teile von Saros/E in unabhängige Komponente extrahiert
 - Darunter auch das Filesystem

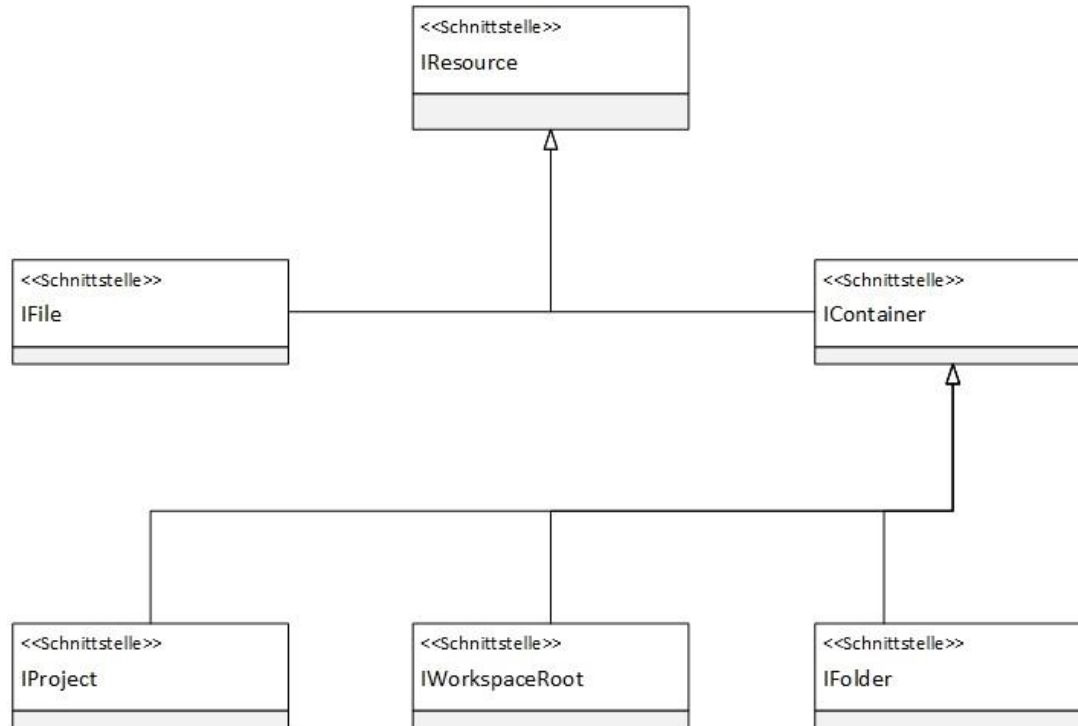
Filesystem Eclipse



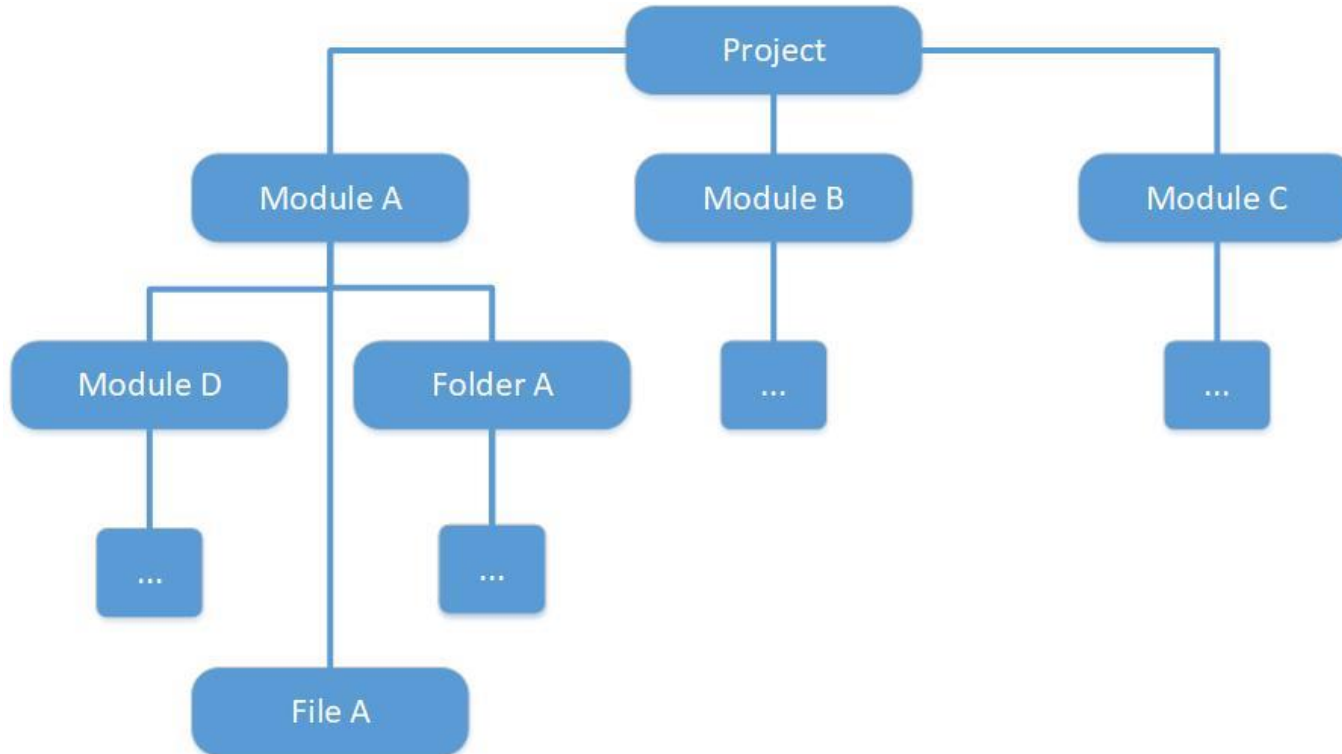
Eclipse: Ressourcenschnittstellen



Saros: Ressourcenschnittstellen



Filesystem IntelliJ



Terminologie: Eclipse und IntelliJ im Vergleich

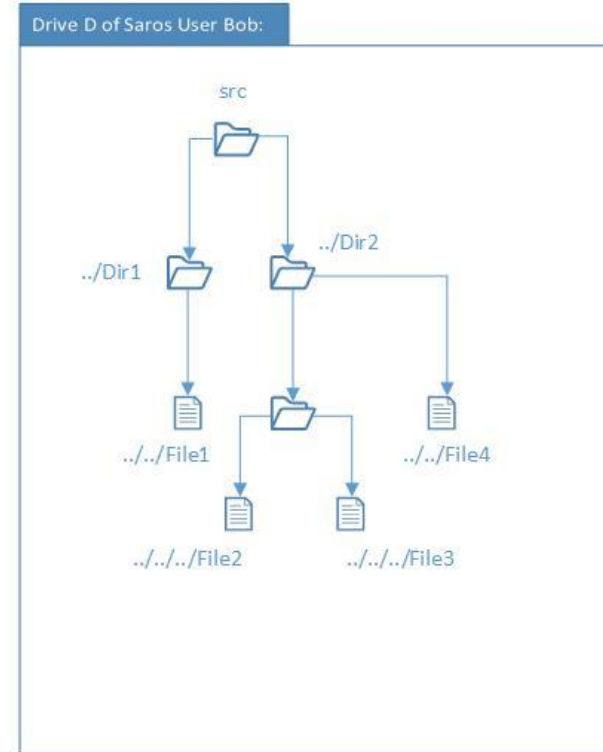
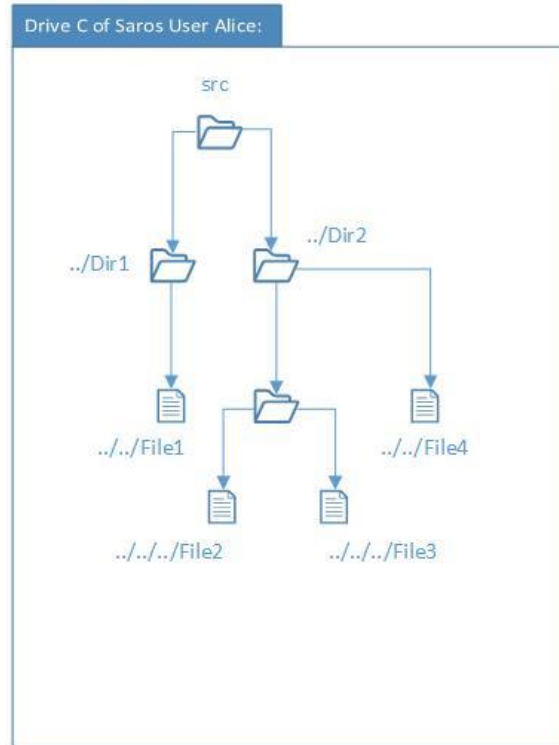
Eclipse	IntelliJ
Workspace	Project
Project	Module
Facet	Facet
Library	Library
JRE	SDK
Classpath variable	Path variable

Quelle: <https://www.jetbrains.com/help/idea/migrating-from-eclipse-to-intellij-idea.html>

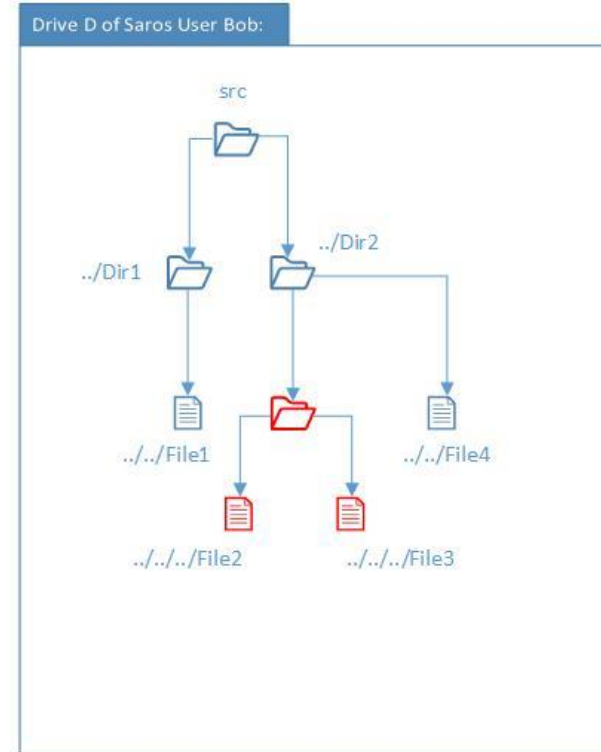
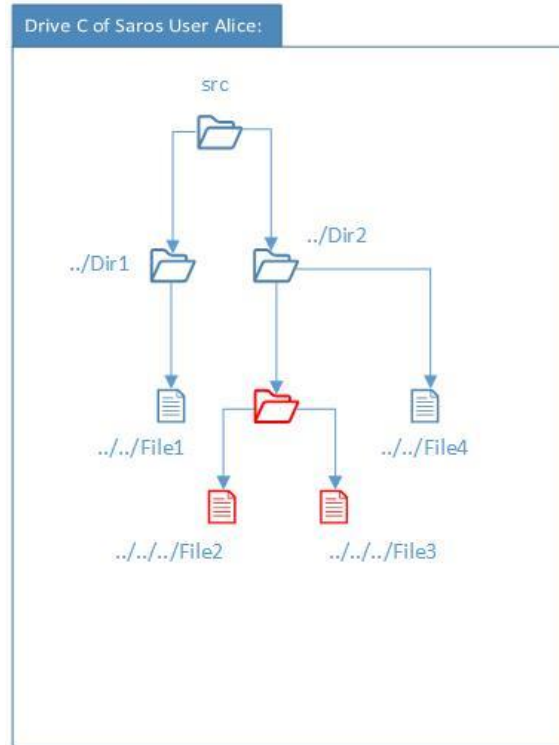
Problem

- Saros Filesystemdesign ähnelt dem von Eclipse zu stark
 - Funktionalitäten von Saros/I eingeschränkt
- Weitere Abstraktion des Saros Filesystems nicht erwünscht

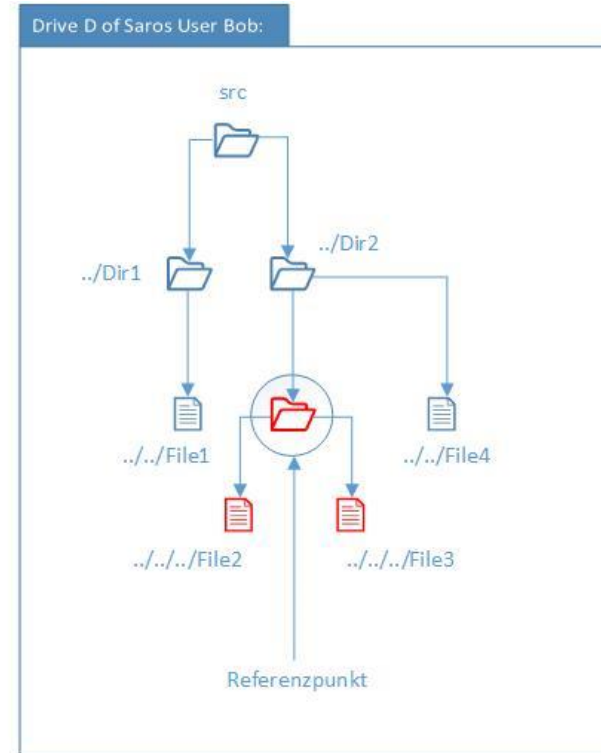
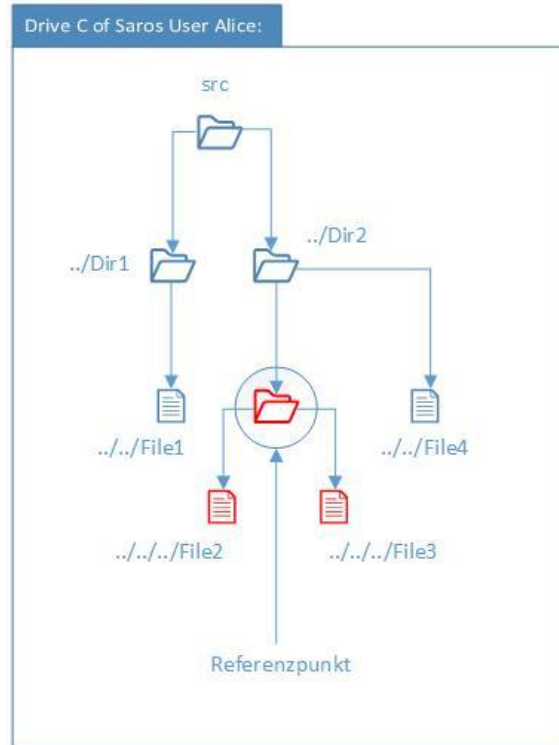
Architektur: Referenzpunkt-basiert



Architektur: Referenzpunkt-basiert

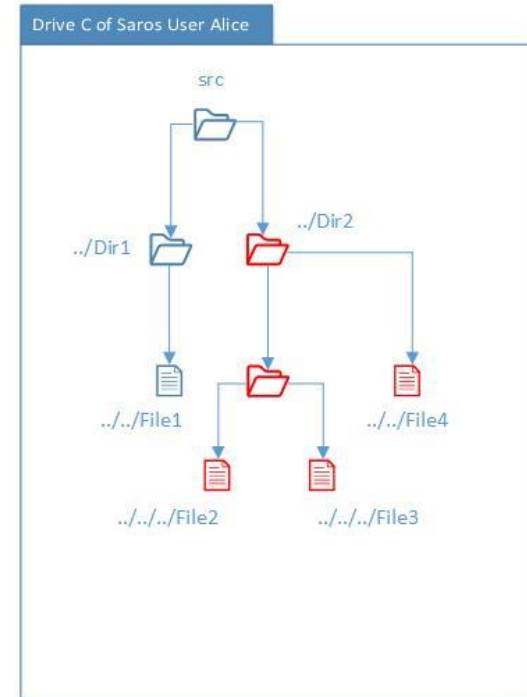


Architektur: Referenzpunkt-basiert



Architektur: Referenzpunkt-basiert

- Ressourcen kennen zunächst seinen Referenzpunkt
- Abgrenzung: Referenzpunkte auf Saros Projekte



Vorgehen/Strategie

- Ermitteln von Saros IProject Referenzen
- Komponentenweises Refaktorisieren
- Duale Architektur
- Neudesign Saros Filesystem

Komponentenweises Refaktorisieren

- Saros Core
- Saros/E
- Saros/I
- Saros Server
- (HTML GUI)

Komponentenweises Refaktorisieren

- Saros Core
- Saros/E
- Saros/I
- Saros Server
- (HTML GUI)
- Strikte Trennung zwischen Komponenten nicht möglich

Duale Architektur

- Projekt- und referenzpunktbasierte Architektur
- Nur während der Refaktorisierung
- Referenzpunktmanager

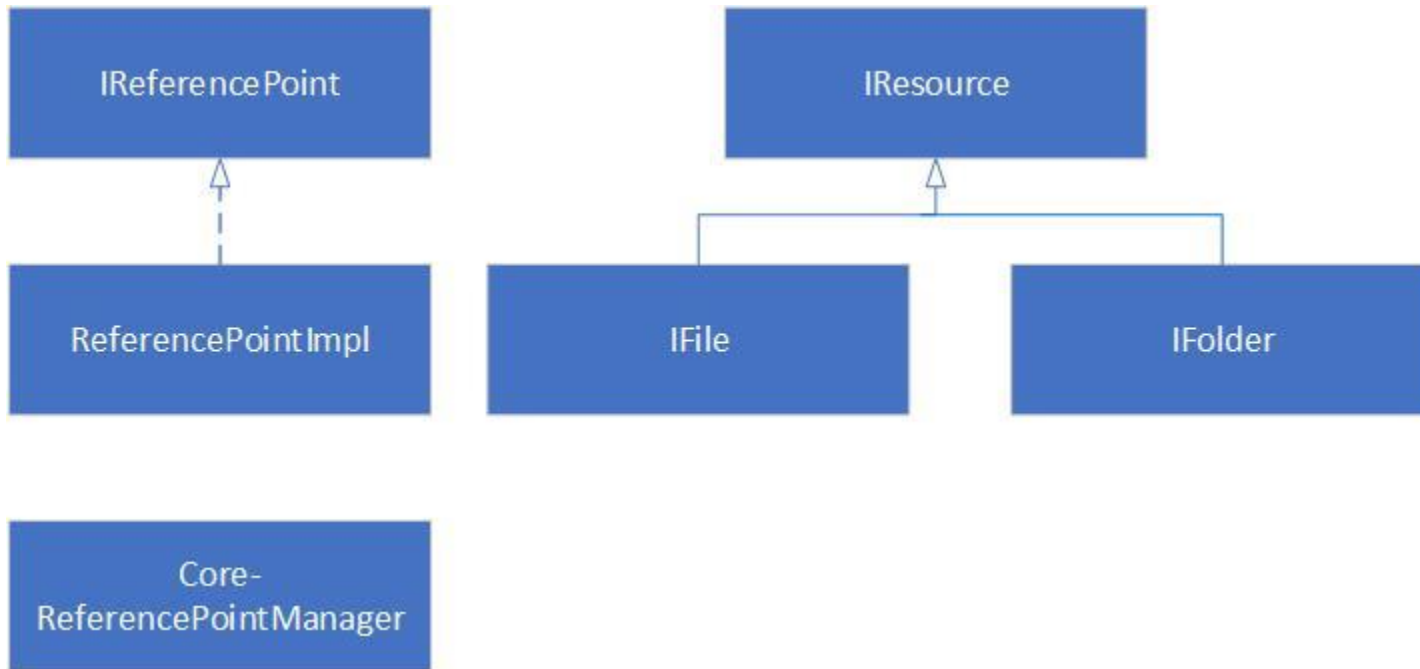
Referenzpunktmanager

- Mappt Referenzpunkte auf Saros Projekte
- Enthält Funktionalitäten von Saros Projekten

Duale Architektur

- 5 Schritte
 1. Project-basierte Schnittstelle als deprecated markieren
 2. Referenzpunktbasierte Schnittstelle implementieren
 3. Bestehende Schnittstelle auf neue verweisen
 4. Komponenten, die auf referenzpunktbasierte Schnittstelle verweisen, auf die neue verweisen
 5. Bestehende Schnittstelle löschen

Design Filesystem Endzustand



CoreReferencePointManager

- Mappt Referenzpunkte auf Saros Folder

IReferencePoint



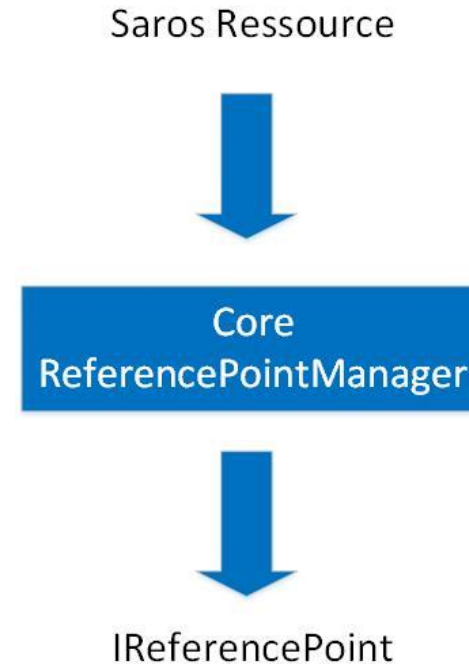
Core
ReferencePointManager



Saros Folder

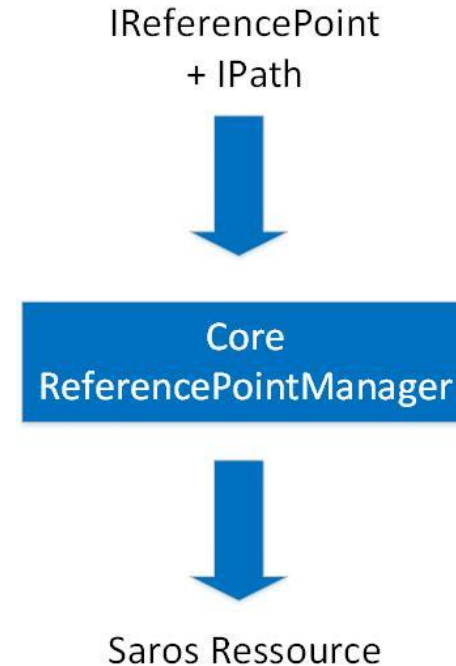
CoreReferencePointManager

- Mappt Referenzpunkte auf Saros Folder
- Erzeugt Referenzpunkte aus Saros Ressourcen

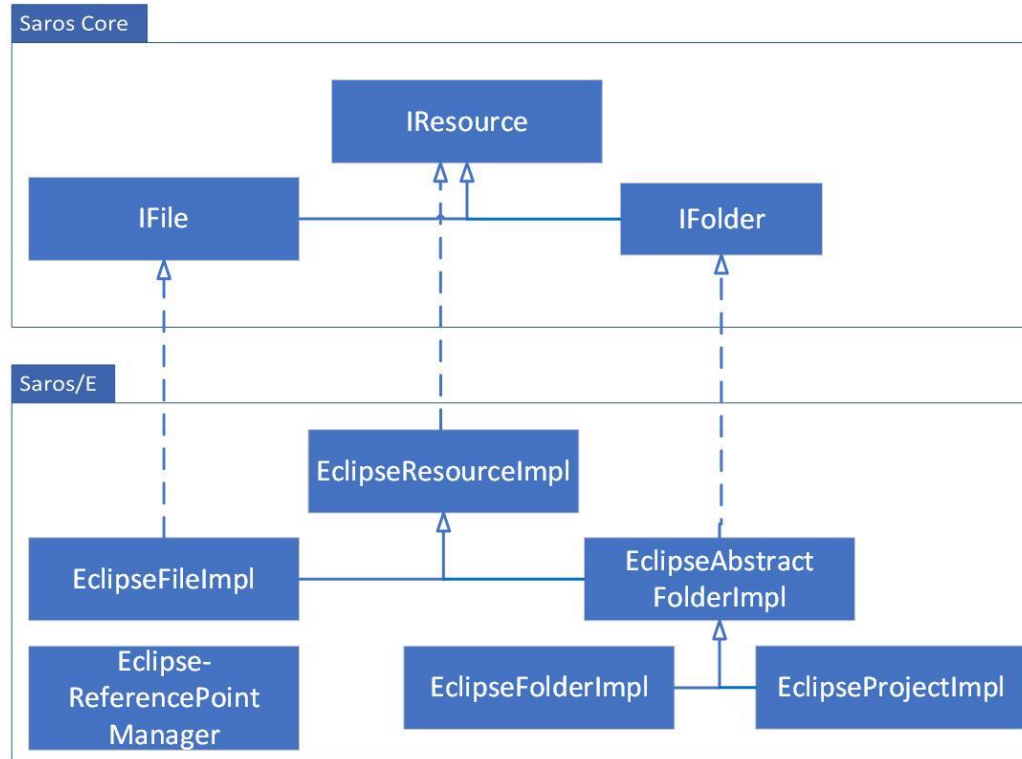


CoreReferencePointManager

- Mappt Referenzpunkte auf Saros Folder
- Erzeugt Referenzpunkte aus Saros Ressourcen
- Ermittelt Saros Ressourcen



Filesystem (Saros/E)



EclipseReferencePointManager

- Mappt Referenzpunkte auf Eclipse Projekte

IReferencePoint



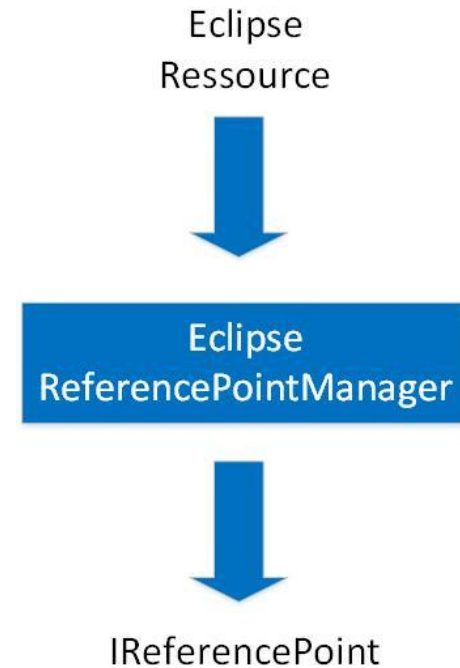
Eclipse
ReferencePointManager



Eclipse Project

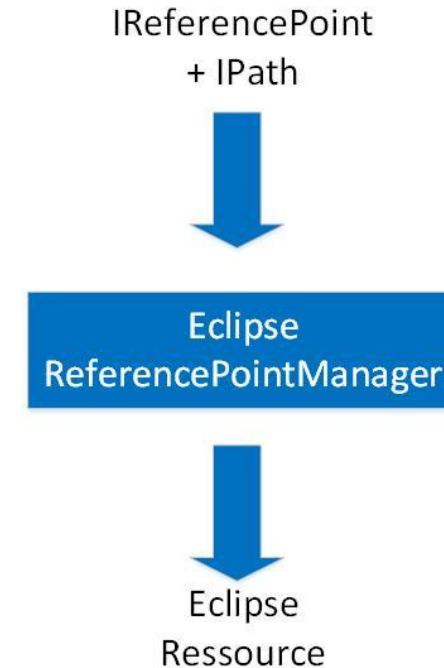
EclipseReferencePointManager

- Mappt Referenzpunkte auf Eclipse Projekte
- Erzeugt Referenzpunkte aus Eclipse Ressourcen

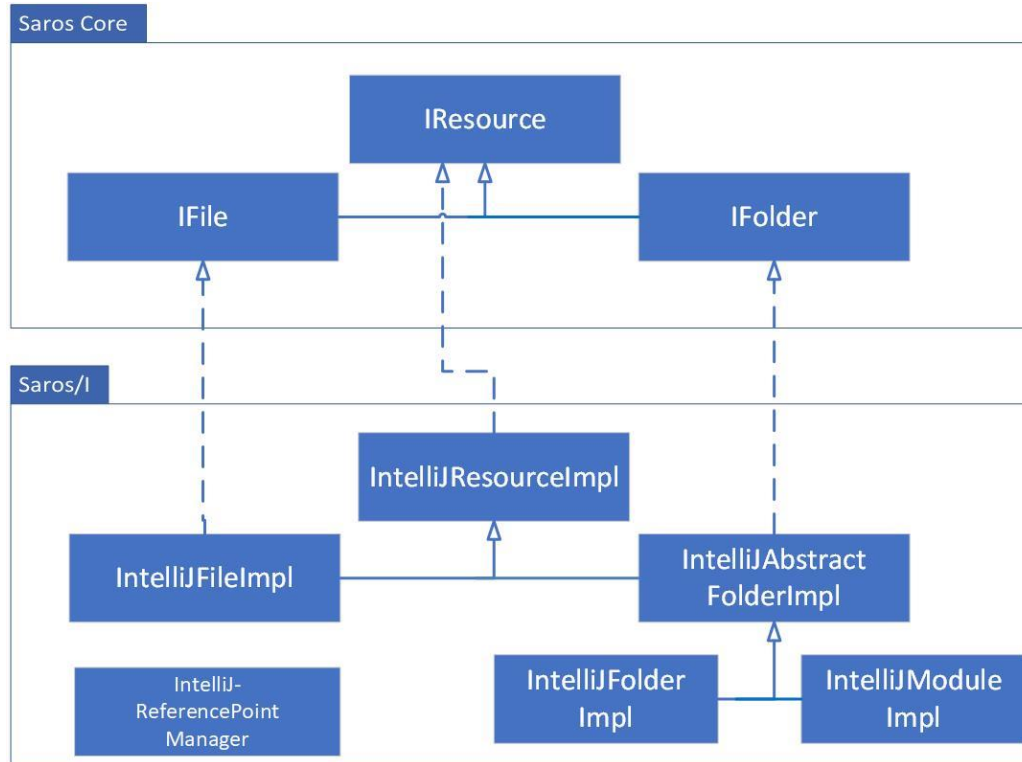


EclipseReferencePointManager

- Mappt Referenzpunkte auf Eclipse Projekte
- Erzeugt Referenzpunkte aus Eclipse Ressourcen
- Ermittelt EclipseRessourcen



Filesystem (Saros/I)



IntelliJReferencePointManager

- Mappt Referenzpunkte auf IntelliJ Module

IReferencePoint



IntelliJ
ReferencePointManager



IntelliJ Module

IntelliJReferencePointManager

- Mappt Referenzpunkte auf IntelliJ Module
- Erzeugt Referenzpunkte aus IntelliJ Ressourcen

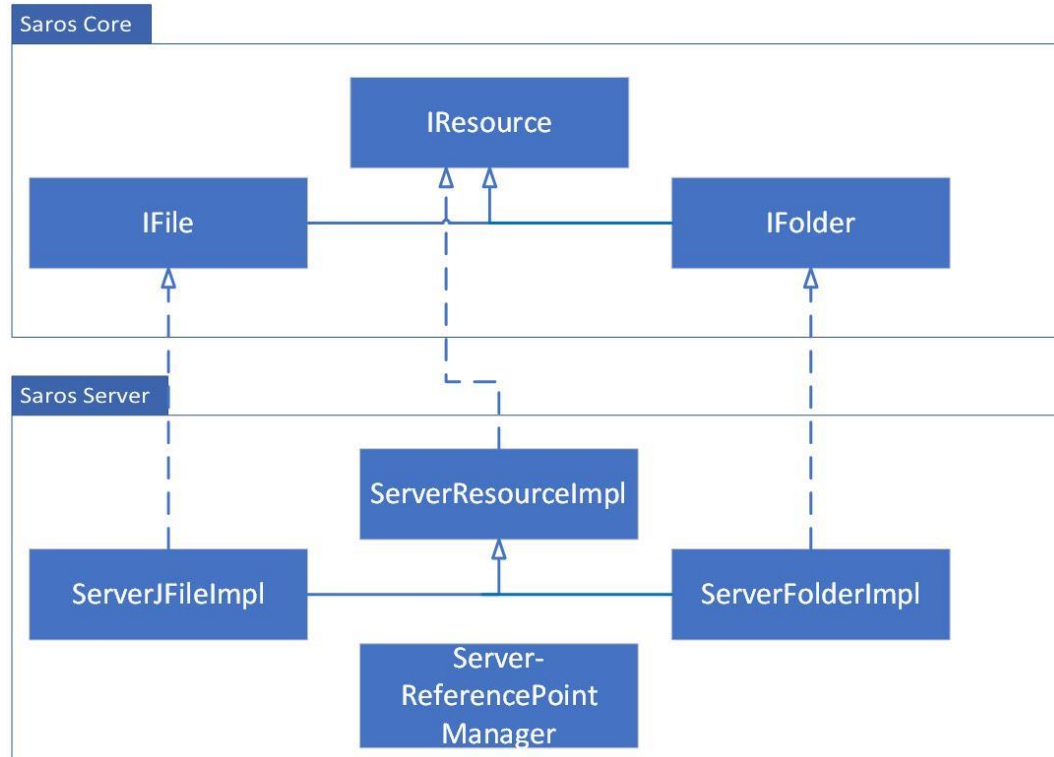


IntelliJReferencePointManager

- Mappt Referenzpunkte auf IntelliJ Module
- Erzeugt Referenzpunkte aus IntelliJ Ressourcen
- Ermittelt IntelliJ Ressourcen



Filesystem (Saros Server)



ServerReferencePointManager

- Mappt Referenzpunkte auf Java IO Files

IReferencePoint



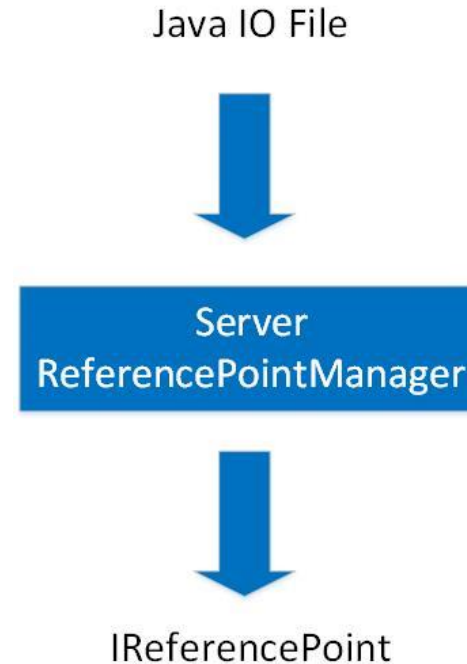
Server
ReferencePointManager



Java IO File

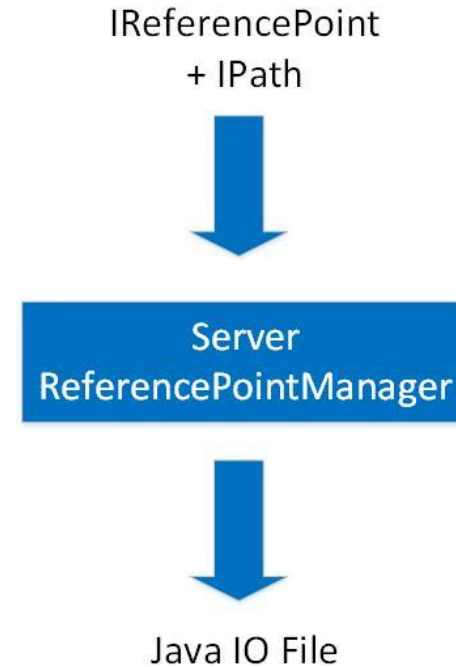
ServerReferencePointManager

- Mappt Referenzpunkte auf Java IO Files
- Erzeugt Referenzpunkte aus Java IO Files



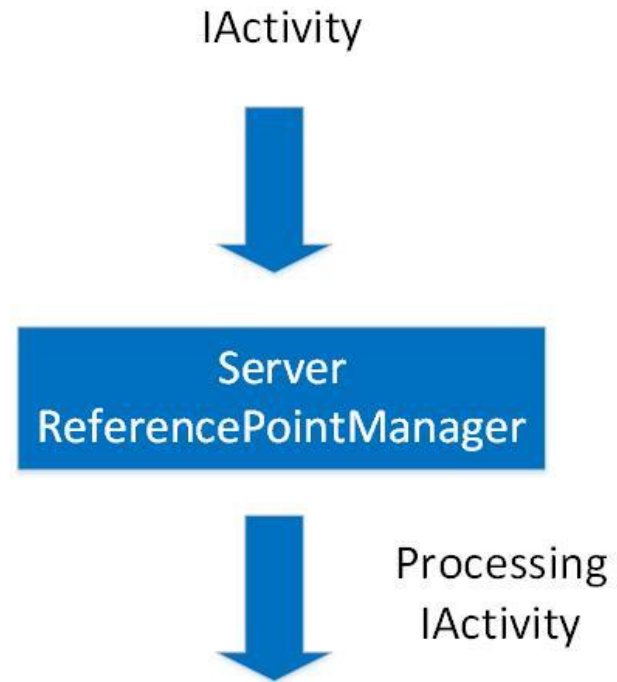
ServerReferencePointManager

- Mappt Referenzpunkte auf Java IO Files
- Erzeugt Referenzpunkte aus Java IO Files
- Ermittelt Java IO Files



ServerReferencePointManager

- Mappt Referenzpunkte auf Java IO Files
- Erzeugt Referenzpunkte aus Java IO Files
- Ermittelt Java IO Files
- Führt Aktivitäten aus



Bewertung

- Proof of Concept konnte gezeigt werden
- Ergebnisse vielversprechend
- Leider kein Code in den Master Branch übernommen

Projekt gescheitert?

- Wurde der Endzustand verstanden?
- Was das Thema zu kompliziert?
- Waren die Prozessschritte falsch?
- Ist der Zielzustand unerreichbar?

Endzustand verstanden?

- Grundannahme: Projektbasierten Design durch Referenzpunktbasiertes Design ersetzen
- Statt Saros Projekte gibt es Referenzpunkte
- Teilprojekte von Saros lösen Referenzpunkte und rel. Pfad nach Ressourcen auf
 - Neudesign der Saros Ressourcen

Prozessschritte falsch?

- Saros Projekt komponentenweise ablösen
- Alle Teilprojekte von Saros betroffen
- Refaktorisierungen teils komponentenübergreifend
- Funktionelle Weiterentwicklung vs. Refaktorisierung

Thema zu kompliziert?

- Kaum Transparenz im Vorgehen/Strategie
 - Gegenmaßnahme: Vorstellung des Endzustandes
- Patches in einem separatem Branch gepflegt
 - Vom Master Branch übernommene Änderungen nicht reviewt
 - Reviewaufwand weiterhin hoch!

Fazit

- Saros auf referenzpunktbasierter Architektur lauffähig
- Änderungen leider nicht übernommen
- Refaktorisierung ist sehr kompliziert
- Lernfaktor für alle sehr hoch

Ende

Vielen Dank für's Zuhören

Fragen?