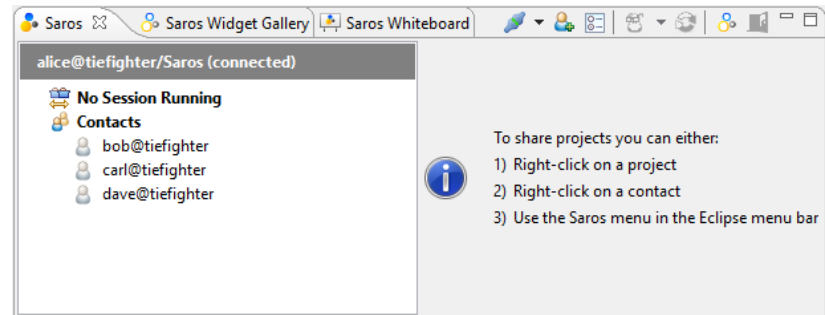


# Antrittsvortrag Masterarbeit

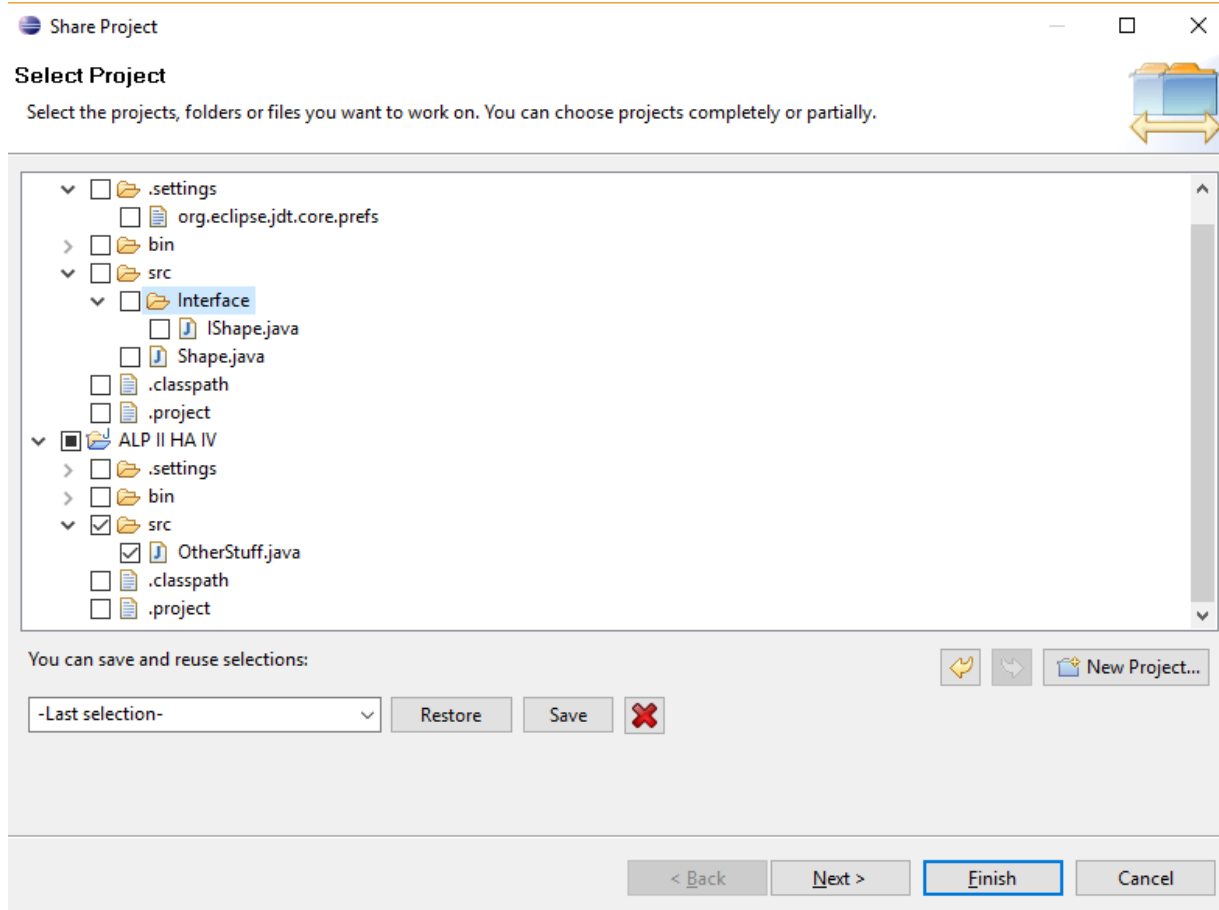
Refactoring Saros Filesystem

# Was ist Saros

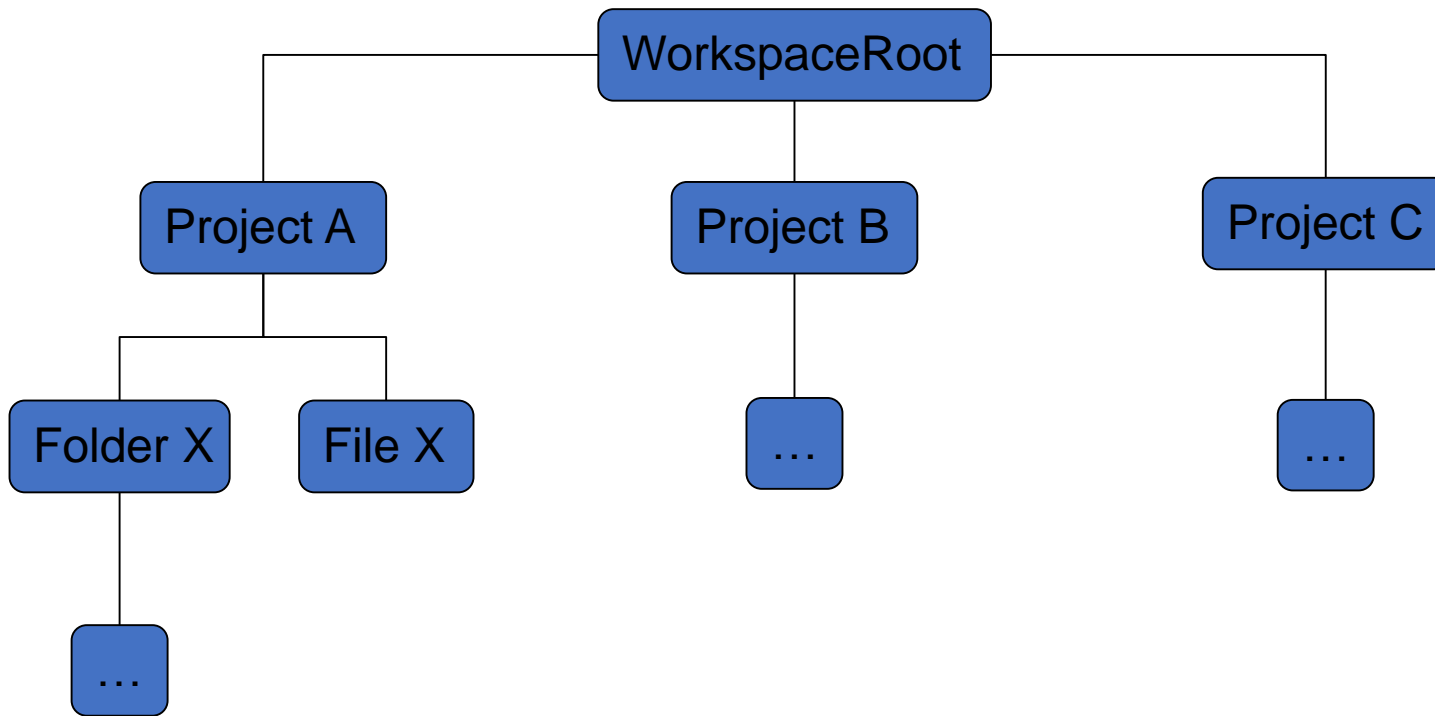
- IDE Plugin für gemeinsame Paarprogrammierung in Echtzeit
- Projekte können geteilt werden



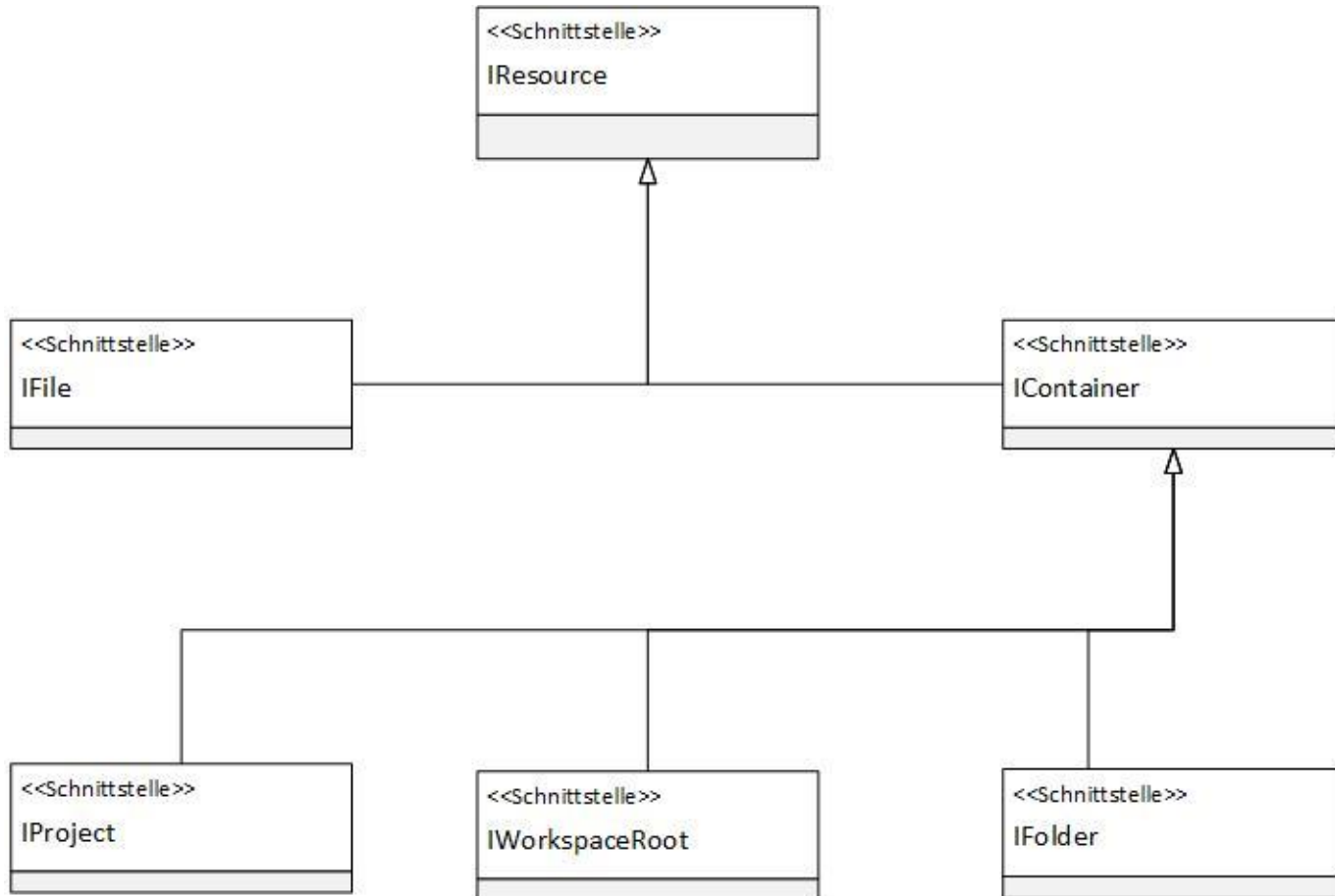
# Was ist Saros



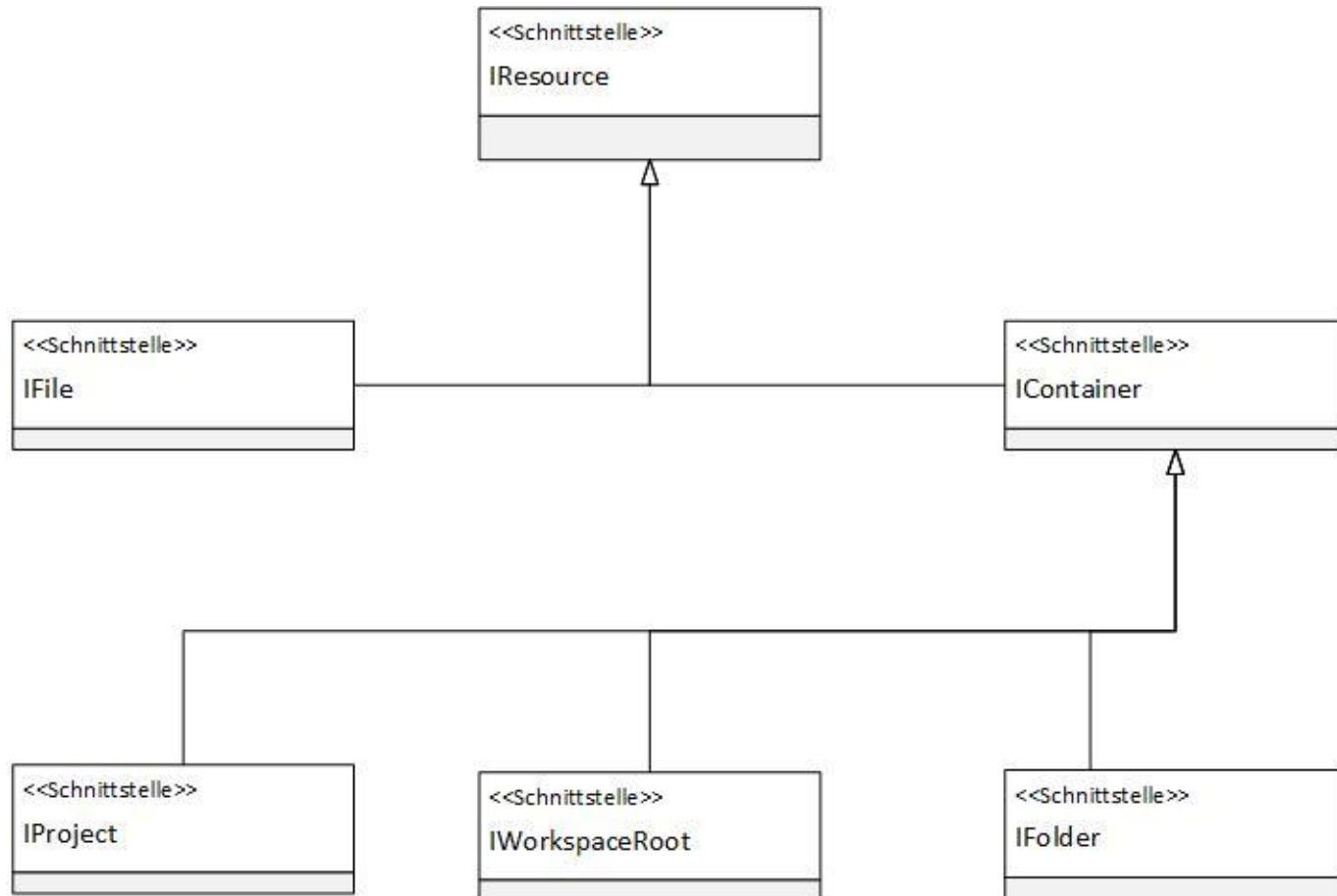
# Eclipse Filesystem



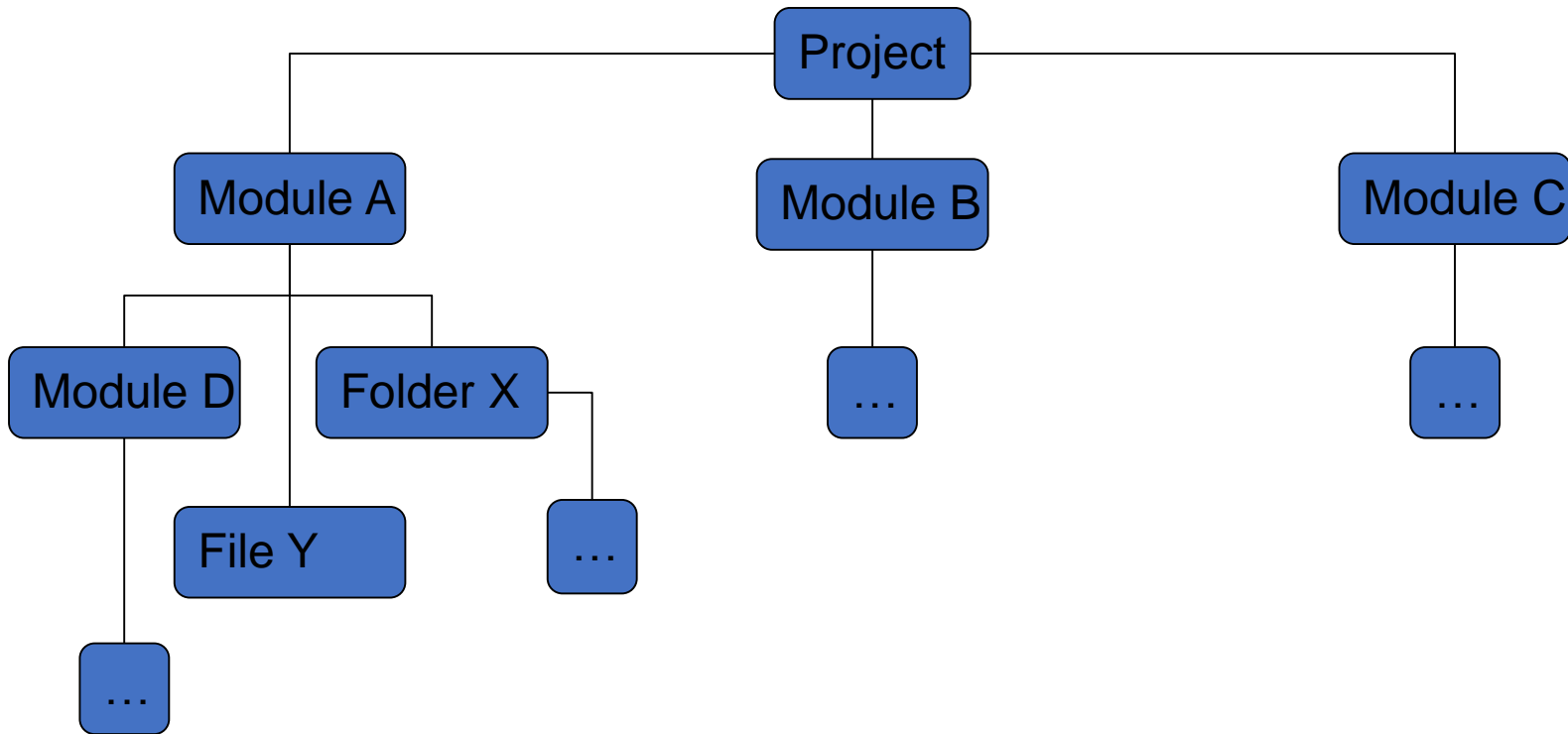
# Eclipse Filesystem



# Saros Core Filesystem



# IntelliJ Filesystem



# Terminologie: Eclipse und IntelliJ im Vergleich

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

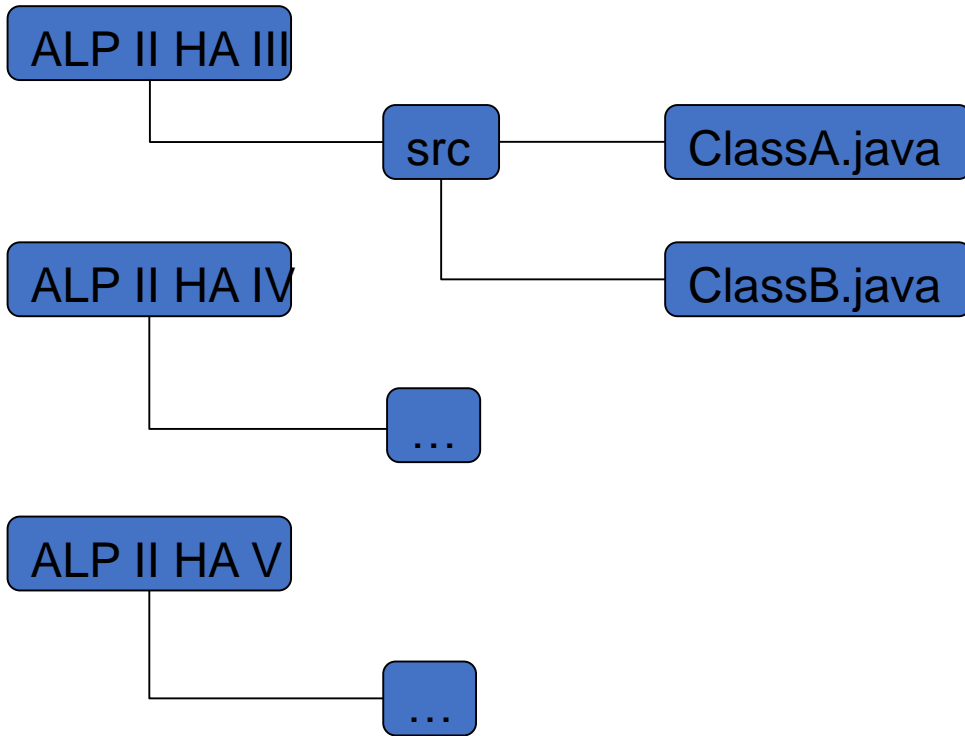
Quelle: <https://www.jetbrains.com/help/idea/migrating-from-eclipse-to-intellij-idea.html>



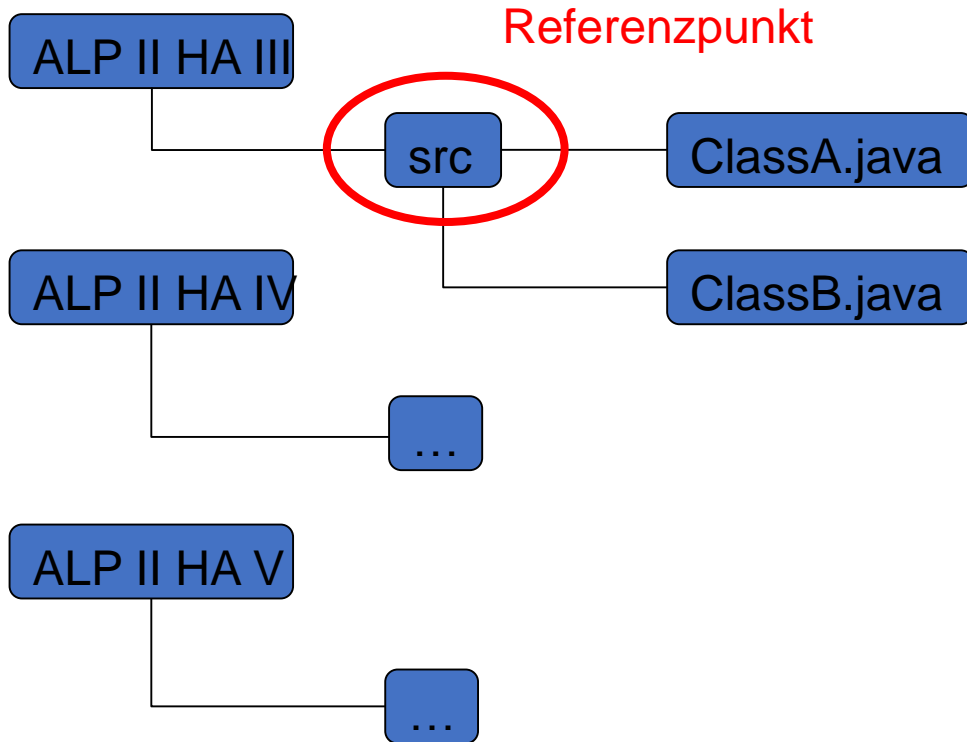
# Schritte des Refactorings (Ziele)

1. Entwurf und Einführung des überarbeiteten Filesystems
2. Ersetzung des projektbasierten Designs des Saros Cores
3. Anpassung Saros/E und Saros/I (und Saros Server)
4. Anpassung der GUI

# Resourcetree

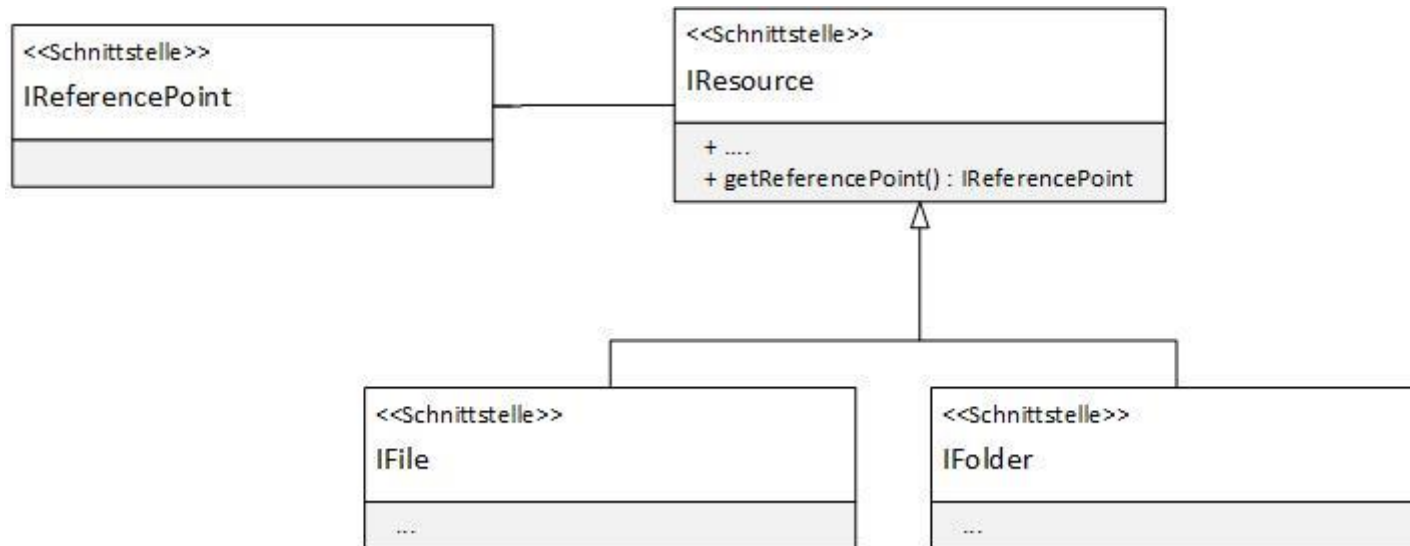


# Resourcetree



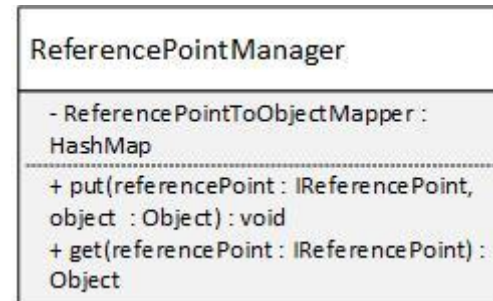
- Verschiedene abs. Pfade
- Gemeinsamkeit: gleicher Referenzpunkt
- Ressourcen kennen ihren Referenzpunkt

# Saros Filesystem (Zielzustand)



# Ersetzung des projektbasierten Designs des Saros Cores

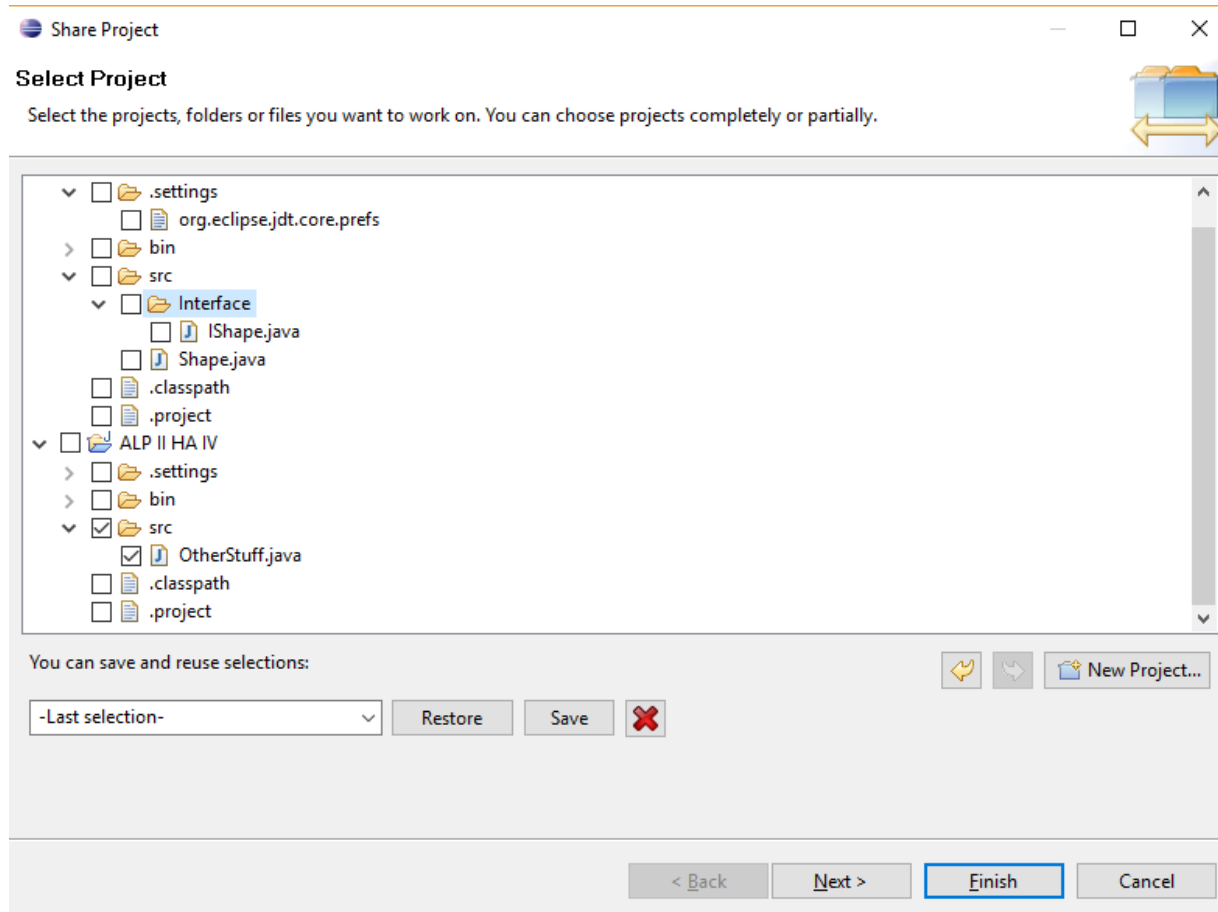
- Saros Core Komponenten  
iterativ nach dem neuen  
Filesystem anpassen
- Zuständigkeiten von Saros  
Projekt ermitteln und  
berücksichtigen
- Einführung  
IReferencePointManager



# Anpassung Saros/E und Saros/I

- Die letzten Saros Projekte werden ersetzt
- Einführung und Implementierung von Eclipse- und IntelliJReferencePointManager
  - EclipseReferencePointManager: Mapping von IReferencePoint auf eclipse.Container
  - IntelliJReferencePointManager: Mapping von IReferencePoint auf VirtualFile
- Saros Filesystem Komponenten werden obsolet: IProject, IContainer und IWorkspaceRoot

# Anpassung der GUI



# Offene Fragen/Probleme

- Saros Server?
- IntelliJ Filesystem: Was gibt es hier zu beachten?
- Sollen Referenzpunkte von Saros User selbst gewählt werden?
- Referenzpunkte für Saros User sichtbar?