



Live Service Composition

Philipp Ewen

Jan Zipfler

Inhaltsübersicht

- Theoretisches
- Implementierung
- Live Demonstration



Theoretisches

Oder, was muss man eigentlich darüber wissen?

Begriffserklärung: Service Composition

Choreografie

- Keine Erstellung eines neuen Services
- Definiert Regeln und Richtlinien für die Zusammenarbeit von Services
- Jeder Service sieht nur den eigenen Kontext
- Service trägt nur einen Teil zum gesamten Prozess bei

Orchestrierung

- Erstellung eines neuen Services
- Neue Service übernimmt die zentrale Steuerung

Begriffserklärung

- Registry
 - Enthält Liste aller Services
 - Zuordnung: Servicename → Service
- Service Contract
 - Benötigte Parameter
 - Rückgabewerte
 - Beschreibung
- Service
 - Umsetzung einer für sich selbst stehenden fachlichen Funktionalität.
- Execution Engine
 - ausführen von zusammengesetzten Services

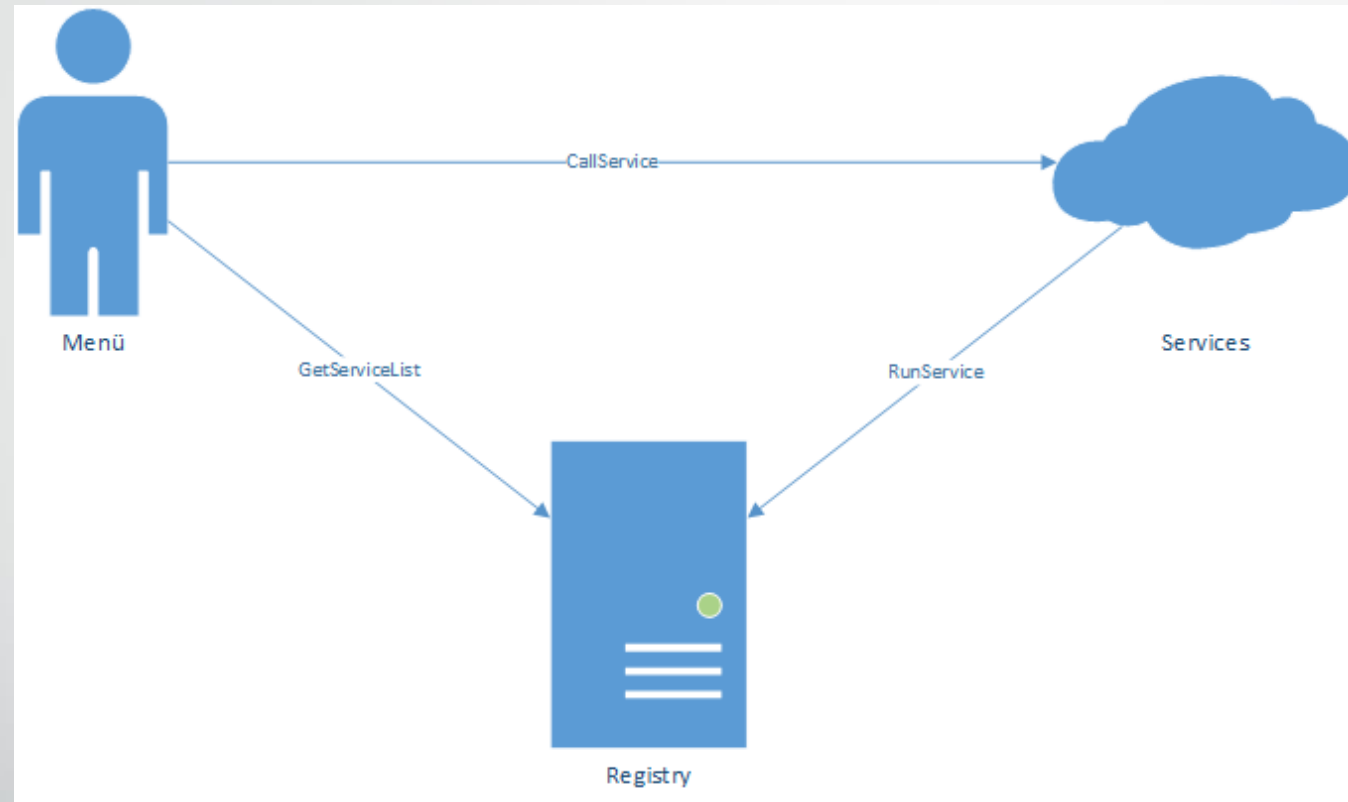


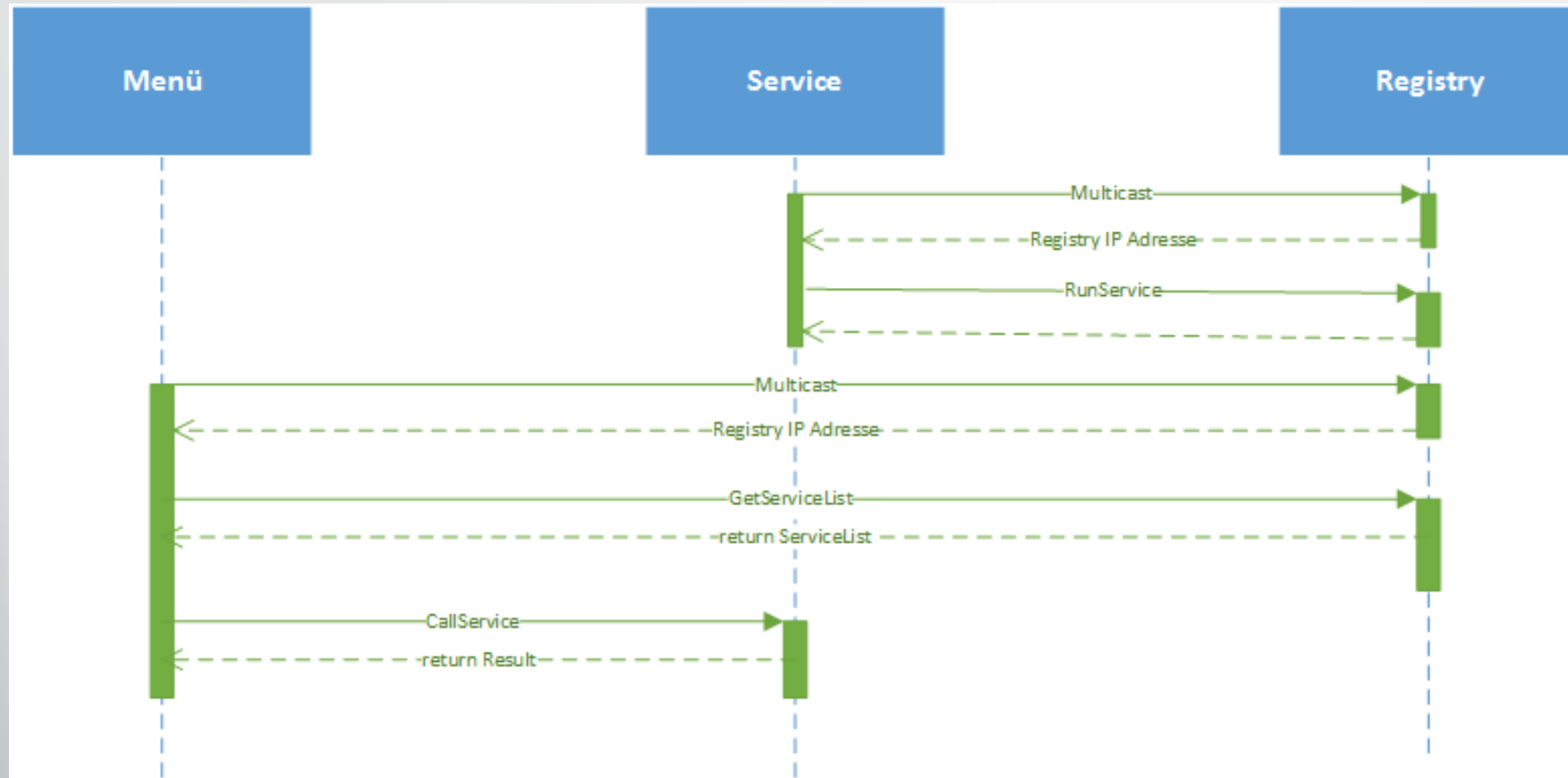
Implementierung

Was wurde wie implementiert?

Grundsätzliches


- GO als verwendete Programmiersprache
 - Entwickelt von Google
- GIT als Versionskontrollsystem
- Erweiterbares Konzept
 - und Realisierung
- Registry erreichbar über Multicast





Beispiel: Definition eines Services

```
1 // Service Contract definieren
2 serviceRandom := service.ServiceInfo{
3     "random",
4     "int",
5     "Generates a random integer.",
6     []service.ArgumentInfo{
7         {"void", "void", "no arguments"},
8     },
9 }
10
11 // ...
12
13 // Hauptfunktion des Services
14 func randomHandler(servicecall *service.ServiceCall) string {
15     number := rand.Int()
16
17     return strconv.Itoa(number)
18 }
19
20 // ...
21
22 // Service registrieren und ausführen
23 service.RunService(&serviceRandom, randomHandler)
```



Live Demonstration



Vielen Dank für Ihre Aufmerksamkeit

Und einen schönen Feierabend.