

Folie: „Überblick“

Erläutern.

Folie: „UDP-Port“

- Daten versenden
- Daten empfangen
- In unserem Fall: Von Haskell bereitgestellt

Folie: „MessageParser“

- Wendet Infrastrukturdaten an
- Nachrichten \rightarrow interne Datenstrukturen
- Signaturen überprüfen

Folie: „Distributor“

- Verteilung
 - Nachrichten an lokale Nutzer \leftrightarrow Nachrichten für externe Empfänger
 - Aktualisierung der Daten über das Netz: ROUTING, CHANNEL, JOIN, LEAVE, CERTIFICATE in Data
- Packt OBSCURE aus
- Beantwortet GETCERTIFICATE, GETKEY

Folie: „ServerInterface und GUI“

- Trennung der lokalen Node und der lokalen Nutzer
- Definierte Schnittstelle, Datenaustausch über Ereignisse
- Vorerst: direkte Aufrufe, später Möglichkeit mehrere Benutzer anzuschließen

Folie: „Data“

- Aktuelle Routingtabelle
- Alle bekannten Channel mit ihren Benutzern
- Alle bekannten Zertifikate
- Lokale Benutzerinfos
 - Lokale Benutzer des Nodes
 - Channel, in denen die Nutzer sind
 - Zertifikate

Folie: „Transmitter“

- Sendet Broadcasts: Channel-Announcements, ROUTINGs, HELLOs.
- Routet Nachrichten, wenn erforderlich
- Generiert XML-Dokumente, die verschickt werden

Folie: „Geschäftsprozess“

- **Geschäftsprozess:** Nachricht an öffentlichen Kanal
- Beispielhaft einen Geschäftsprozess mit Szenario, Grob-analyse, Feinanalyse.

Folie: „Szenario“

- **„User joint öffentlichen Kanal“**: Node 4 bekommt das JOIN nicht mit
- Node 4 „lernt“ durch Channel-Announcement
- **„Nachricht wird geroutet“**: Zwei Alternativen: Erfolg und Misserfolg

Folie: „Grobanalyse“

Behandeln der Nachricht auf einem Node erklären.

Folie: „Feinanalyse“

Detaillierte Aufrufe erklären.

Folie: „Verteilungsentwurf“

- Nicht aussagekräftig
- **aber ...**
- Verteilung von Nodes und GUIs erklären.

Folie: „Wiederverwendbare Komponenten“

- MessageParser, Transmitter: Umwandlung XML \leftrightarrow interne Formate
- ServerInterface, GUI: Wegen Abstraktion

Folie: „Fragen“