

# Vortrag III: Abschluss Entwicklung einer GUI für den gMix-Simulator

Malte Weinschenk, Jörg Langnickel,  
Jan Carsten Lohmüller & Alexander Beifuß

27. Januar 2014

# Inhalt

- 1 Einleitung
- 2 GUI
- 3 Architektur
- 4 Annotations
- 5 Dependency Checker
- 6 Live Demo

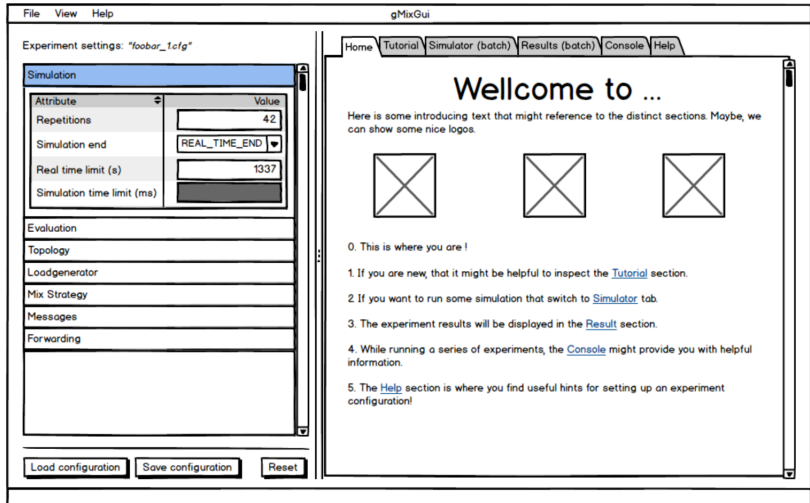
# Ausgangssituation

# Motivation

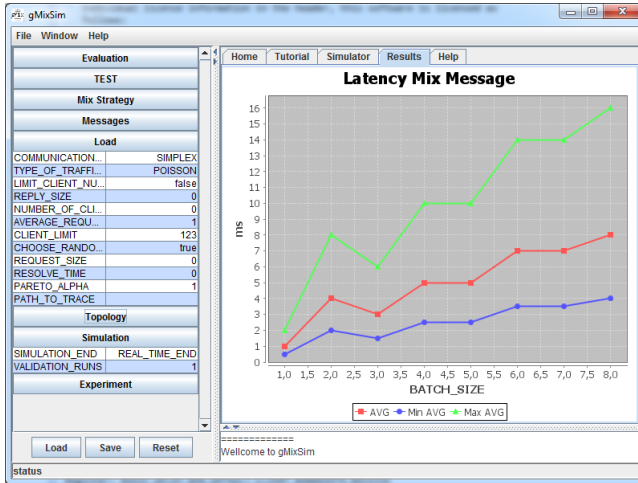
# Benutzergruppen

# Ziele

# Designidee



# Erste Umsetzung





# gMixGUI

File Edit Window Help

### Plugin Configuration

- ☒ Underlay-net
- ☐ Topology
- ☒ Mix Server
- DISTINCT\_USER\_BATCH
- Maximum reply delay: 100
- ☒ Mix Proxy
- ☒ Recoding Scheme
- ☒ Mix Client
- ☒ Load Generator

### Properties To Vary

---

### General Configuration

- ☒ Simulation
- ☒ Plottype

Load Save Reset

### Configuration Selection

```
etc/experiments/example_old.cfg
etc/experiments/experiment.cfg
etc/experiments/new.cfg
```

\* Multiple selection is possible

### Simulation Control

Start Simulation

Stop Simulation

Clear Results

### Export Results

Export Graph

### Simulation Status

### Console

```
2014-01-23 21:16:00 Item in property not found: CONSOLEFRAME_XPOS
2014-01-23 21:16:00 Using default value for item CONSOLEFRAME_XPOS
2014-01-23 21:16:00 Item in property not found: CONSOLEFRAME_YPOS
2014-01-23 21:16:00 Using default value for item CONSOLEFRAME_YPOS
2014-01-23 21:16:00 Item in property not found: CONSOLEFRAME_WIDTH
2014-01-23 21:16:00 Using default value for item CONSOLEFRAME_WIDTH
2014-01-23 21:16:00 Item in property not found: CONSOLEFRAME_HEIGHT
2014-01-23 21:16:00 Using default value for item CONSOLEFRAME_HEIGHT
2014-01-23 21:16:01 Item in property not found: GUISERVICE_TOGGLE_HOME_TAB
2014-01-23 21:16:01 Using default value for item GUISERVICE_TOGGLE_HOME_TAB
2014-01-23 21:16:01 Can not read value for LSB_REPLY_RATE
2014-01-23 21:17:11 Finished simulator with results
```

### Results

Experiment 0

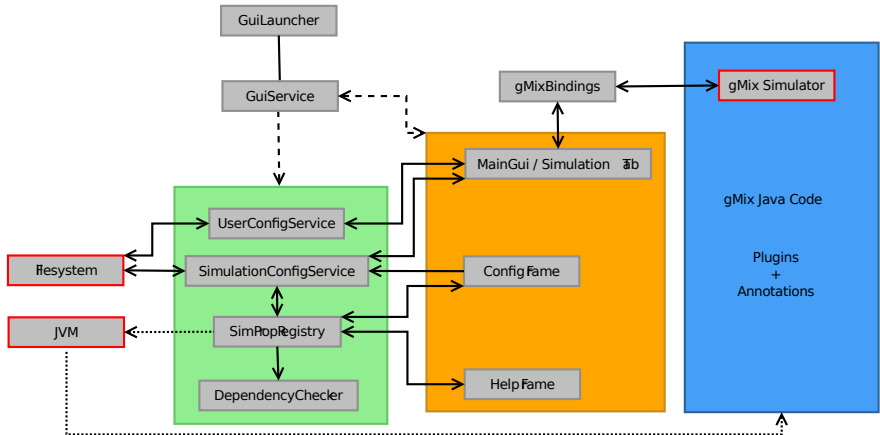
plot\_clientLatencyMixMessage - effect of BATCH\_SIZE

ms

BATCH\_SIZE

MEDIAN AVG (AVG\_CLIENT\_LATENCY\_REQUESTMESSAGE) — +  
 MIN AVG (AVG\_CLIENT\_LATENCY\_REQUESTMESSAGE) — +  
 MAX AVG (AVG\_CLIENT\_LATENCY\_REQUESTMESSAGE) — +  
 MEDIAN MAX (MAX\_CLIENT\_LATENCY\_REQUESTMESSAGE) — x  
 MIN MAX (MAX\_CLIENT\_LATENCY\_REQUESTMESSAGE) — x  
 MAX MAX (MAX\_CLIENT\_LATENCY\_REQUESTMESSAGE) — x

# Architektur



# Motivation für die Verwendung von Annotations

	Annotations	XML zentr.	XML dezent.	Statisch
Plugin Struktur	++	-	+	--
Initialer Aufwand	-	++	+	++
Aufwand neues Plugin	++	+	+	--
Erweiterbarkeit Fkt.	+	+(+)	+(+)	?
Unterstützung d. IDE	++	+	+	++

Für GUI-Benutzer ist die verwendete Technik transparent.

Plugin-Entwickler profitieren jedoch sehr von den Annotations.

⇒ Es sind wenig Gedanken zur GUI nötig.

⇒ Rapid Prototyping

Der Mehraufwand bei der Programmierung des Frameworks ist gerechtfertigt.

## Anwendungsbeispiel für Annotationen

```
@PluginSuperclass(  
    layerName = "Underlay-net",  
    layerKey = "TYPE_OF_DELAY_BOX",  
    position = 7)  
public abstract class DelayBoxImpl {
```

```
@Plugin(  
    pluginName = "Basic delay",  
    pluginKey = "BASIC_DELAY_BOX")  
public class BasicDelayBox extends DelayBoxImpl {  
  
    @IntSimulationProperty(  
        name = "Packet Size (byte)",  
        key = "NETWORK_PACKET_PAYLOAD_SIZE")  
    private int packetSize = new ...
```

# Dependency Checker

- Konfigurationsdateien
- Keine Unterstützung von Abhängigkeiten und Wertebereichen
- Abhängigkeiten zwischen Properties:
  - Value Requirements z.B. Minimal- und Maximalwerte
  - Enable Requirements z.B. gegenseitiger Ausschluß
- Wertebereiche in einem Property: Minimal- und Maximalwerte
- Ziele:
  - Einfache Benutzung
  - Maximale Flexibilität
  - Wertebereiche in einem Property beachten

# Live Demo

Live Demo

# Ende

Vielen Dank für Ihre Aufmerksamkeit