

# Halo Live Stream

Marvin Hofmann, Tobias Haslop

# Einleitung

1. Aufgabe
2. Komponenten
  - Verteiler Server
  - Halo Server
  - Client
3. Bedienung
4. Qualität unserer Lösung
5. Projektorganisation
6. Stärken und Verbesserungsbereiche

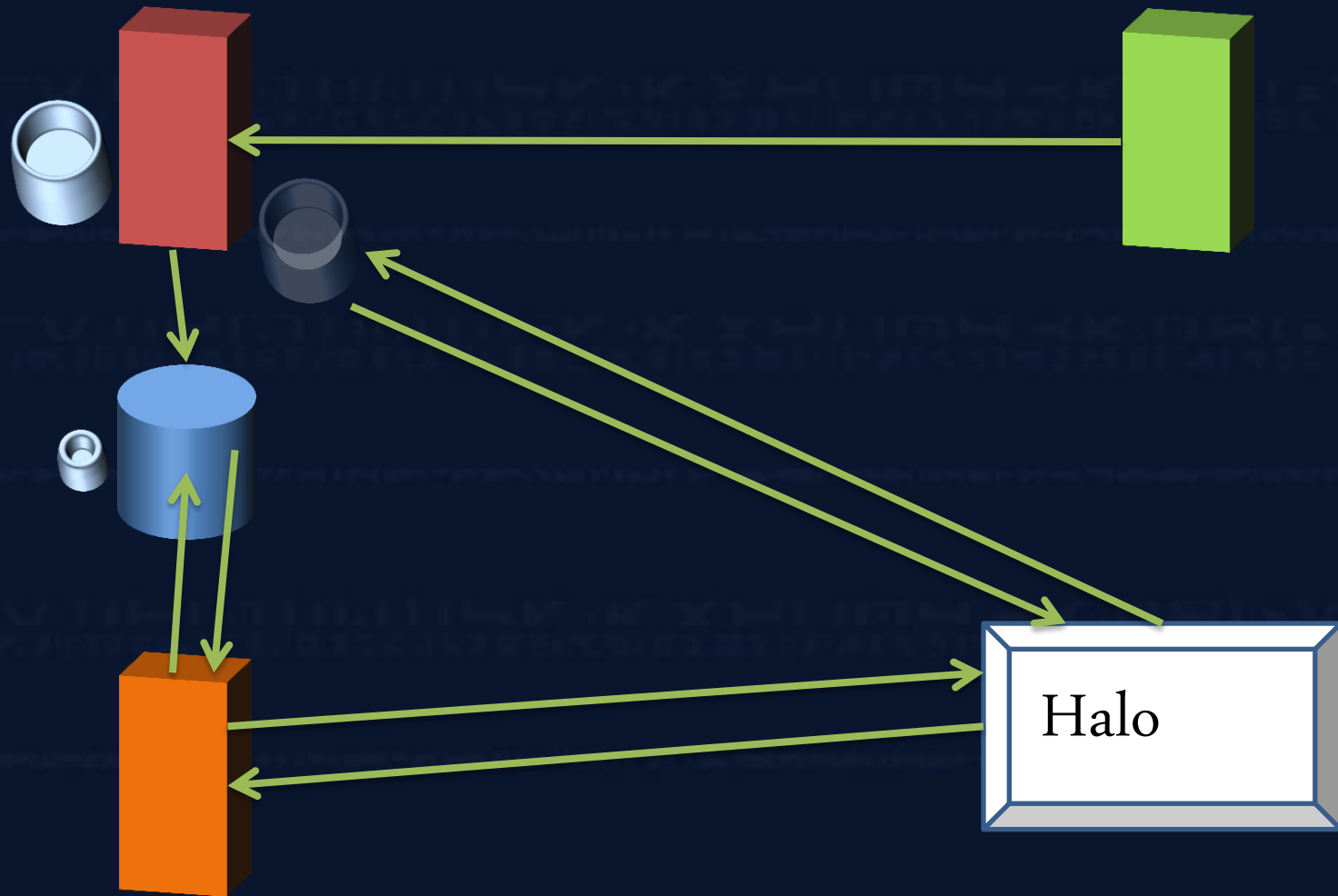
# Aufgabe

Halo: Kampf um die Zukunft

Zuschauermodus + Video-Stream

Ansatz einer Streaming Lösung mit interaktiver  
Kamera

# Verteiler Server



# Halo Server

Servererweiterung Phasor

Daten aus dem Speicher lesen

Socket-Verbindung mit dem Verteiler

Bibliothek LuaSocket

# Client

## Flash - Actionscript 3

- Away3D
  - dreidimensionaler Raum
  - Laden von Körpern, Texturen und Animationen
- Robotlegs
  - Model View Controller Framework

# Spielsituation



# Speichern der Daten

- Alle 33ms einen Spielstand speichern
  - Hoher Speicherverbrauch



- Nur Änderungen speichern
  - Berechnung dauert evtl. zu lange





# Speichern der Daten

- Änderungen und letzten Spielstand speichern
  - Keine beliebigen Zeitsprünge



- Änderungen speichern, sowie Spielstände in größeren Abständen



# Installation

- Scripting-Verzeichnis
  - „%user%/Documents/My Games/Halo/scripts“
- LuaSocket in Server-Root kopieren
- Script starten

# Bedienung



# Demonstration

# Projektorganisation

## Aufgabenverteilung

- Marvin: Verteiler-Server, Datenbank, Website
- Tobias: Client

## Ausnahme Phasorscript:

- Schnittstelle – Marvin
- Erfahrung – Tobias

Projekttag: Schnittstelle zwischen  
Verteiler-Server und Client

# Stärken

- Unabhängige Arbeit
- Verschiedene Programmiersprachen

# Verbesserungsbereiche

- Away3D Entwicklungstempo
- Termin der Schnittstellendefinition
  - Flash Policy File
  - Lange Ladezeiten bei fortgeschrittenem Spiel
  - Optimierungen
- Aufgabenverteilung