Head-Coupled Perspective

Florian Weidner, Sergej Lopatkin

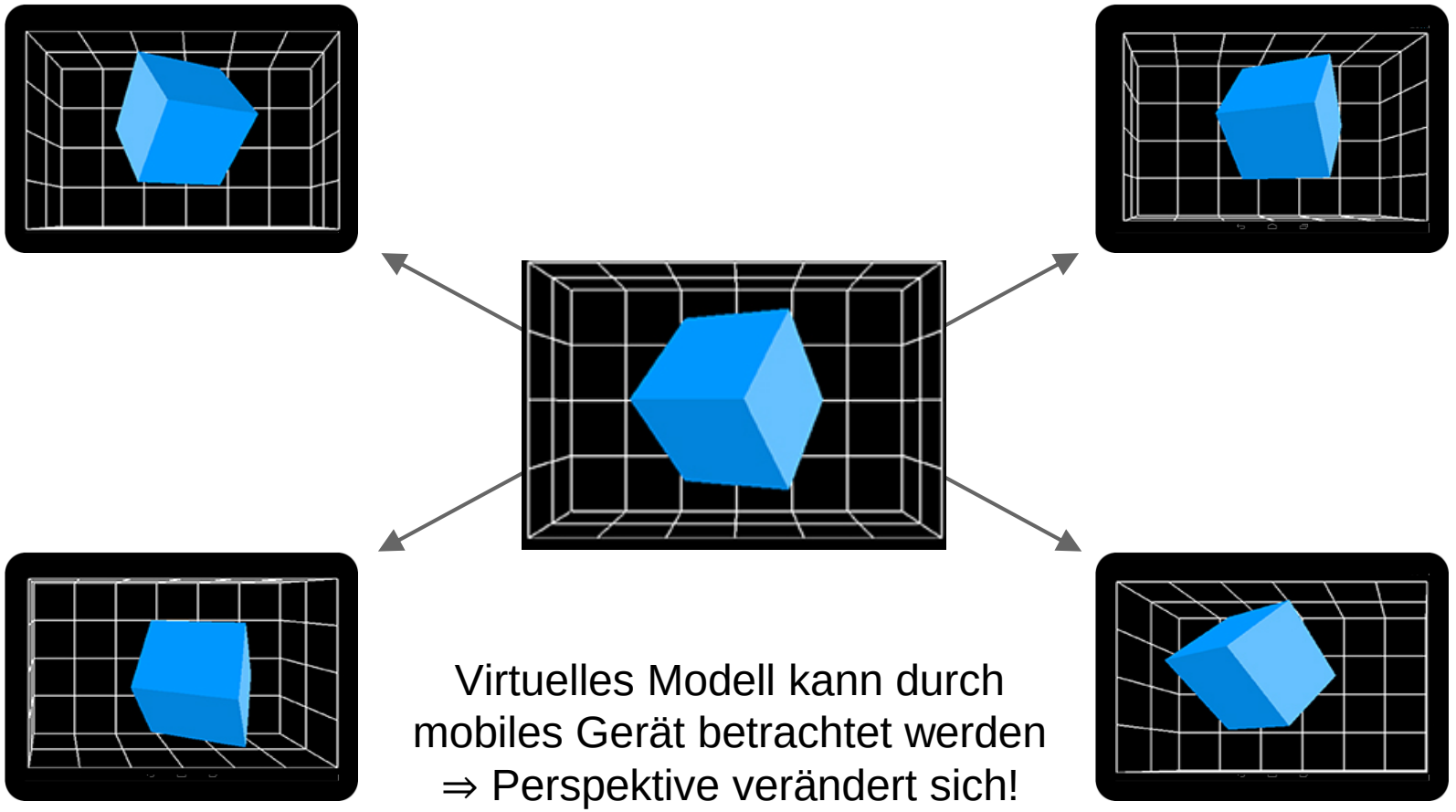
Betreuer: Ricardo Langner, Wolfgang Büschel

TU Dresden, Lehrstuhl MT, WS2013/2014

Gliederung

1. Einführung
2. Zielsetzung
3. Zeitplan
4. Vorgehen
5. Herausforderungen
6. Software
7. Zusammenfassung
8. Fazit

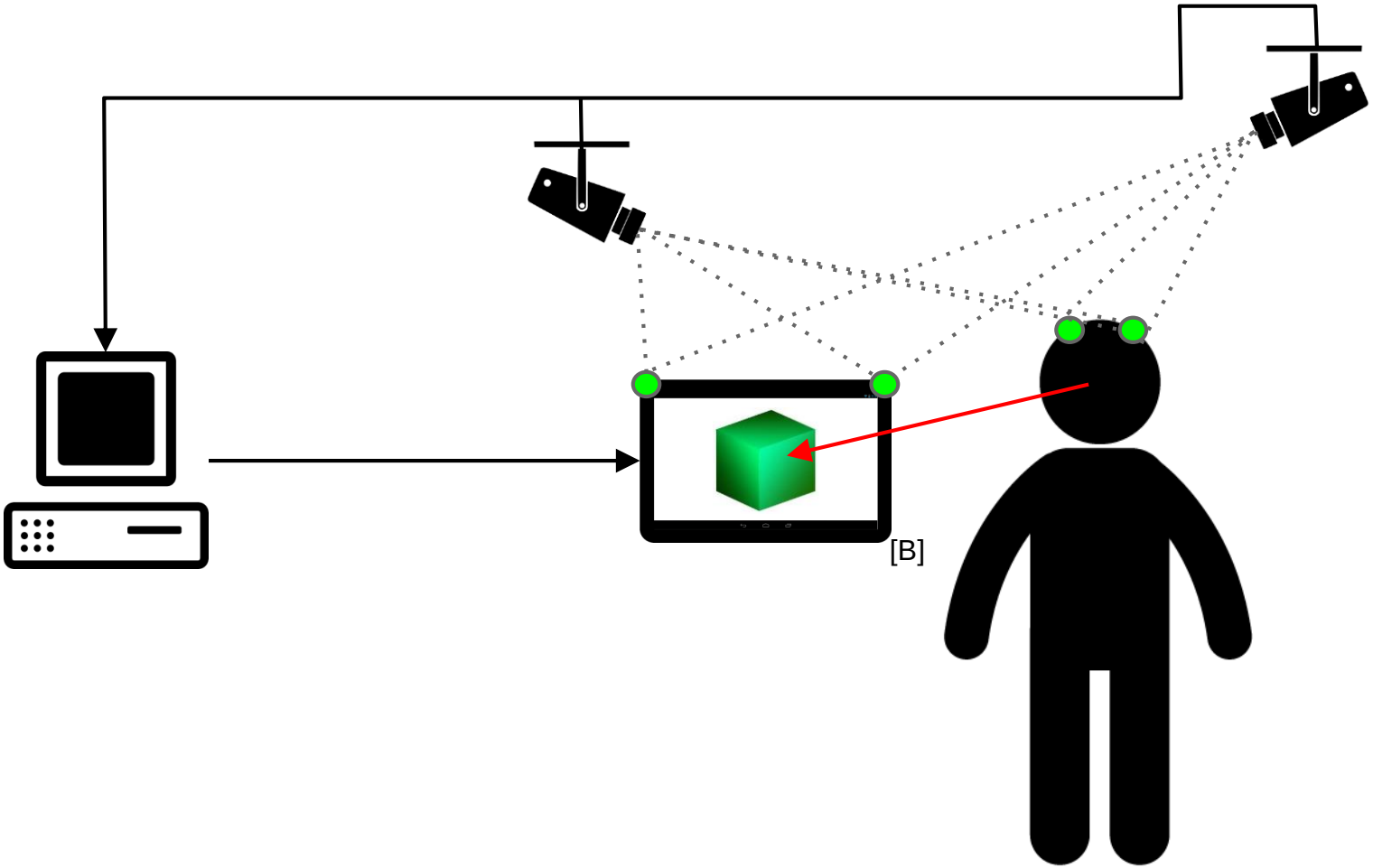
1. Einführung



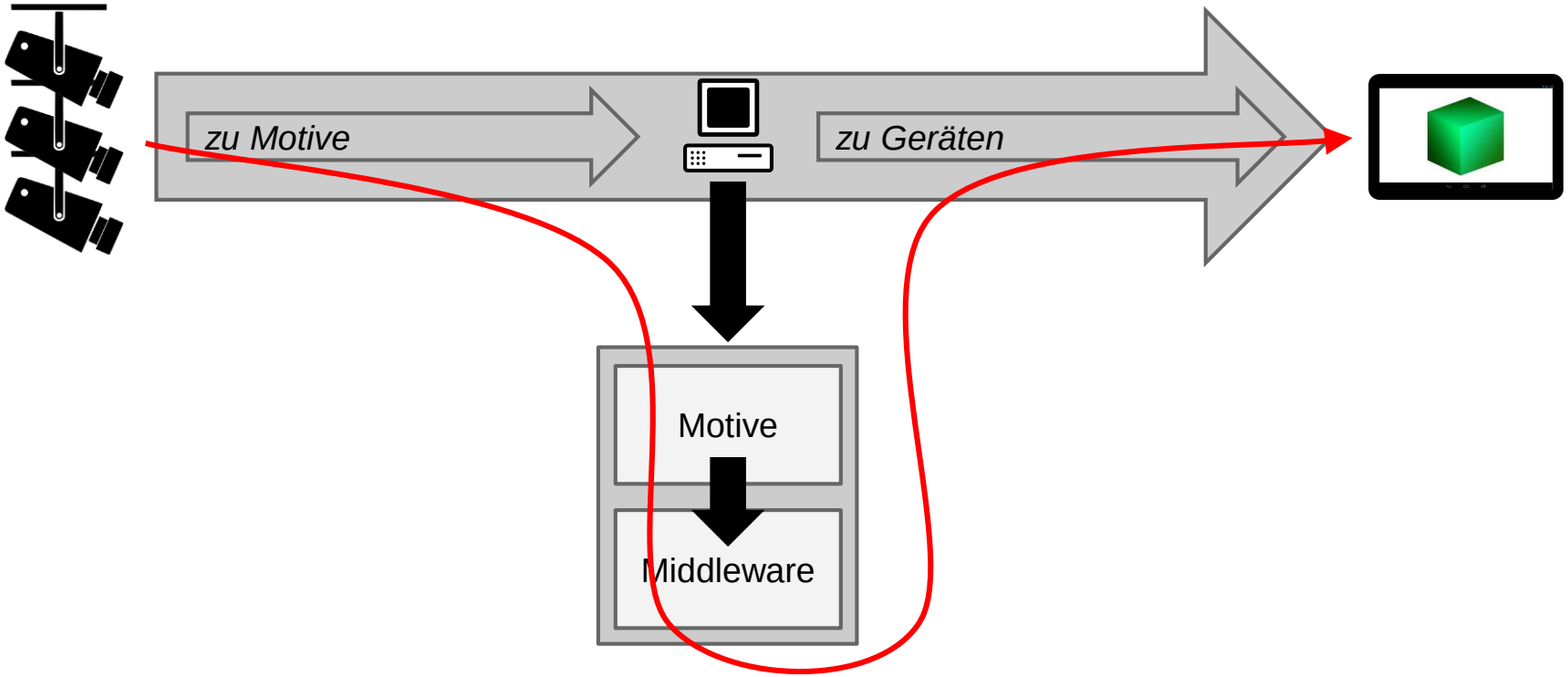
Virtuelles Modell kann durch
mobiles Gerät betrachtet werden
⇒ Perspektive verändert sich!

[A]

1. Einführung



1. Einführung



2. Zielsetzung

⇒ Middleware

- Empfang von Motive
- Verarbeitung
- Senden an Geräte
- Modular/Wiederverwendbar
- Plattformunabhängig

3. Zeitplan

- ✓ Bis 19.11.2013: Einarbeitung
  - ⇒ libgdx, Motive/Streaming, HCP Algorithmus
- ✓ Bis 17.12.2013: Prototypen entwickeln
  - ⇒ Dummy-Daten, Einfache Szene
- ✓ Bis 23.12.2013: Prototypen zusammenfügen
  - ⇒ Daten werden gestreamt und empfangen
- = Bis 21.01.2014: HCP Implementieren
  - ⇒ Datenformat & Verarbeitung, HCP
- = Bis 28.01.2014: Multi-Device Support
  - ⇒ Grundgerüst mit Multi-Device Support in der Middleware
- ✓ Bis 06.02.2014: Doku & Präsentation

4. Vorgehen

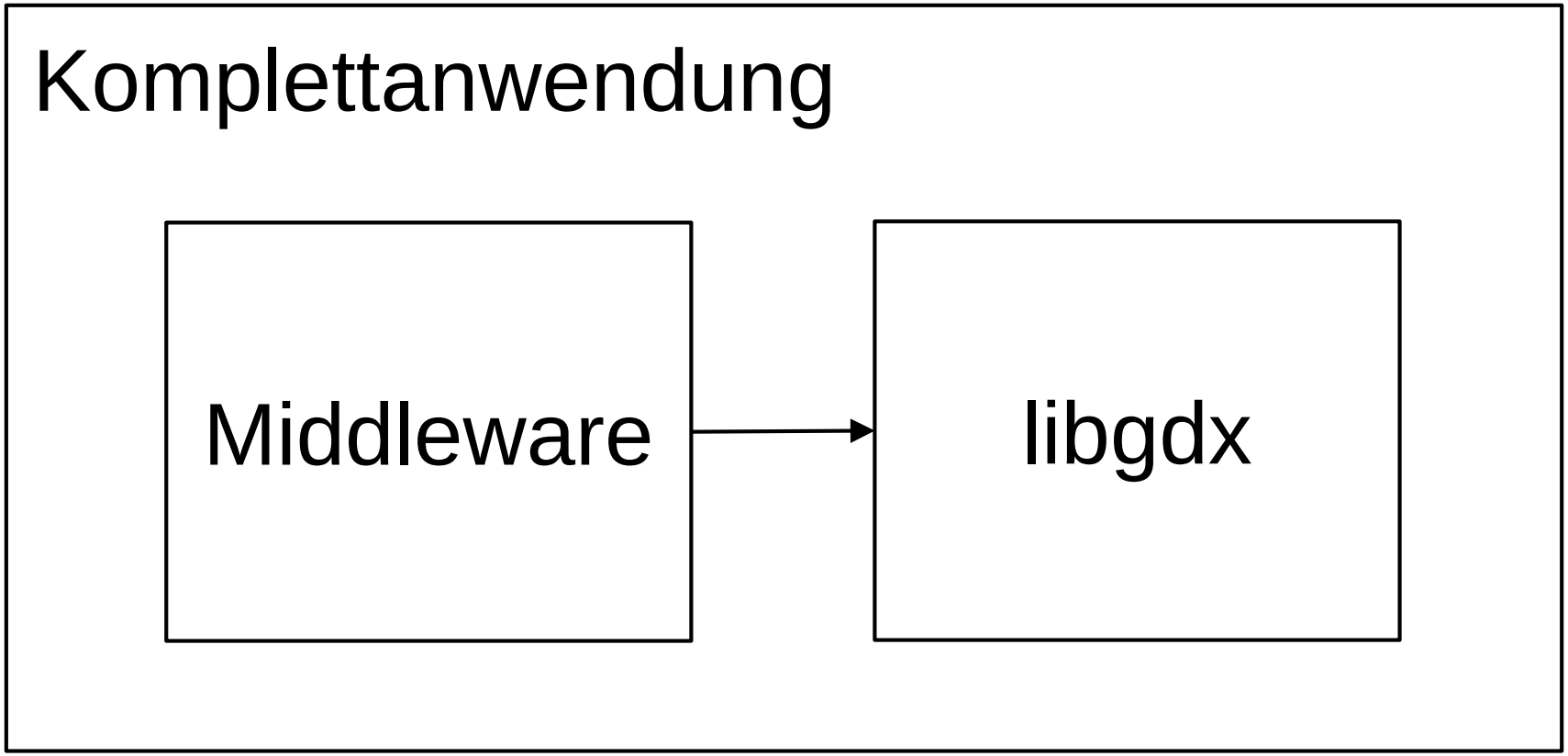
- Parallele Arbeiten:
  1. Florian: Middleware
  2. Sergej: libgdx-Anwendung (3D-Szene)

- Gemeinsame Arbeiten:
  1. Dokumentation/Präsentation
  2. Einarbeitung (Kooima[1], Francone[2])
  3. HCP-Algorithmus (am Ende)

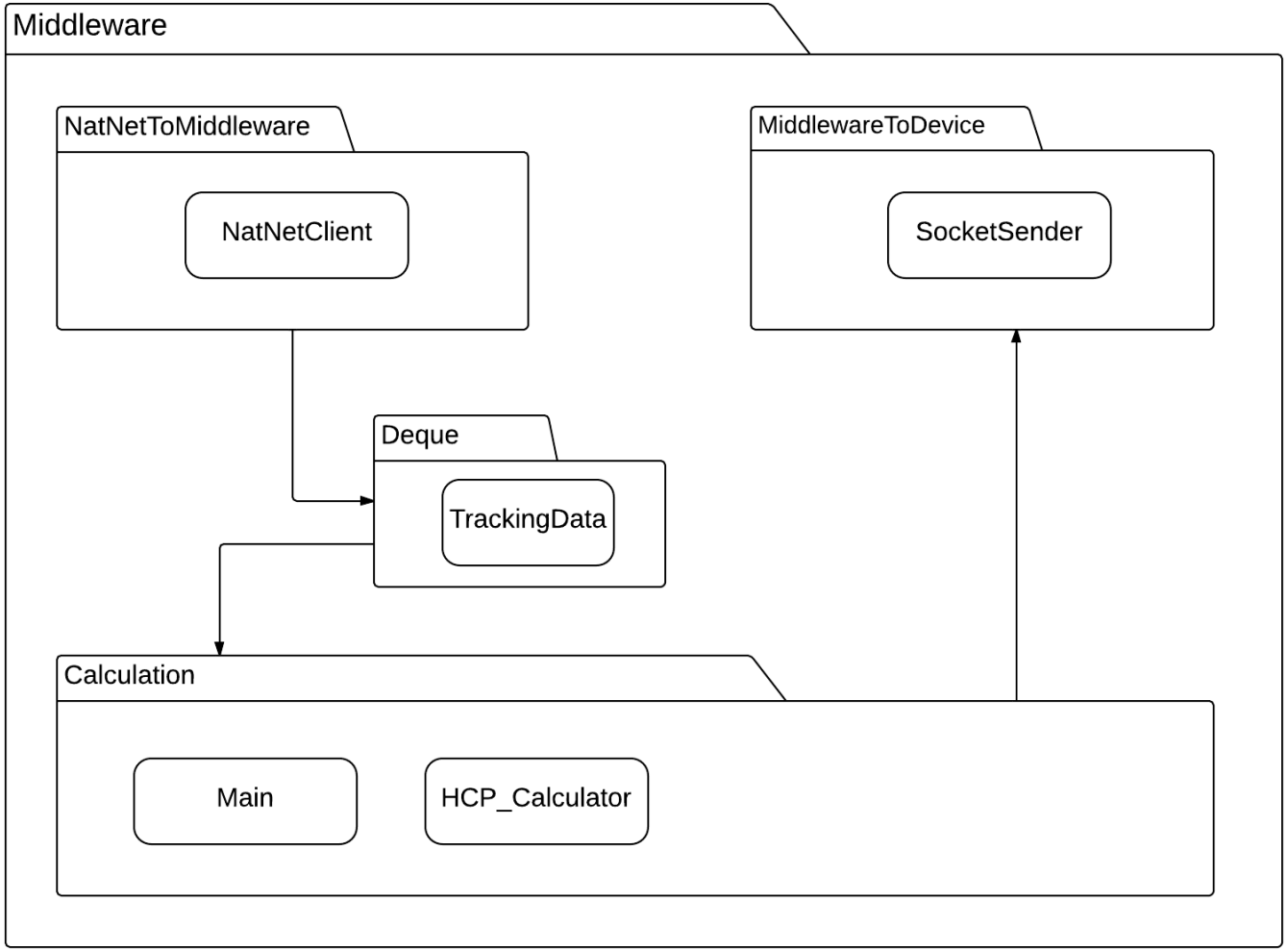
5. Herausforderungen

- Middleware
  - Multithreading in der Middleware
  - Plattformunabhängige Kommunikation
  - Senden, Empfangen, HCP Berechnung
  - Geeignetes Protokoll
- libgdx
  - Szenenerstellung
  - Genauigkeit von float
  - HCP Anwendung!

6. Software: Komplette Anwendung

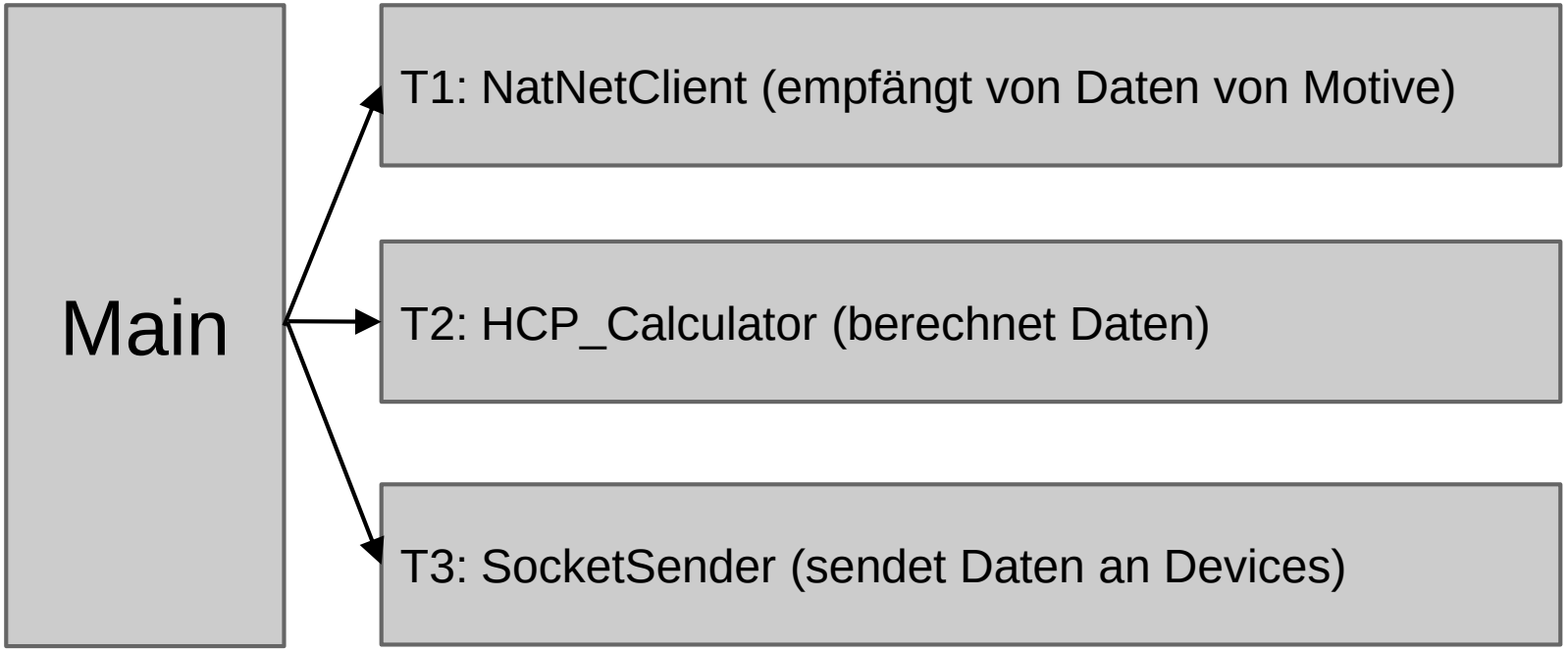


6. Software: Middleware



6. Software: Middleware

Threading



6. Software: Middleware

- Details:
  - ca. 320Byte/Frame mit max 120 Frames/s ⇒ ca. 4kB/s
  - TCP/IP, Socket
- Erstelltes XML-Format:

```
<Response>
  <iFrameNumber></iFrameNumber>
  <head>
    <headID></headID>
    <pivot></pivot>
    <orientation></orientation>
  </head>
  <projMatrix></projMatrix>
  <device>
    <deviceID></deviceID>
    <pivot></pivot>
    <orientation></orientation>
  </device>
</Response>
```

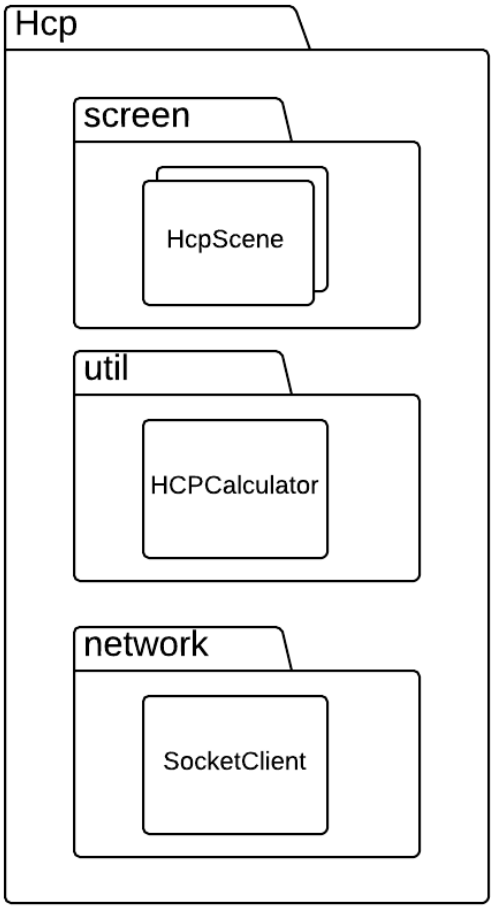
6. Software: Mobile Geräte/libgdx

- Komponenten:
  - Szenen
  - Netwerk
  - HCP Utils

- Trennung der Verantwortlichkeiten

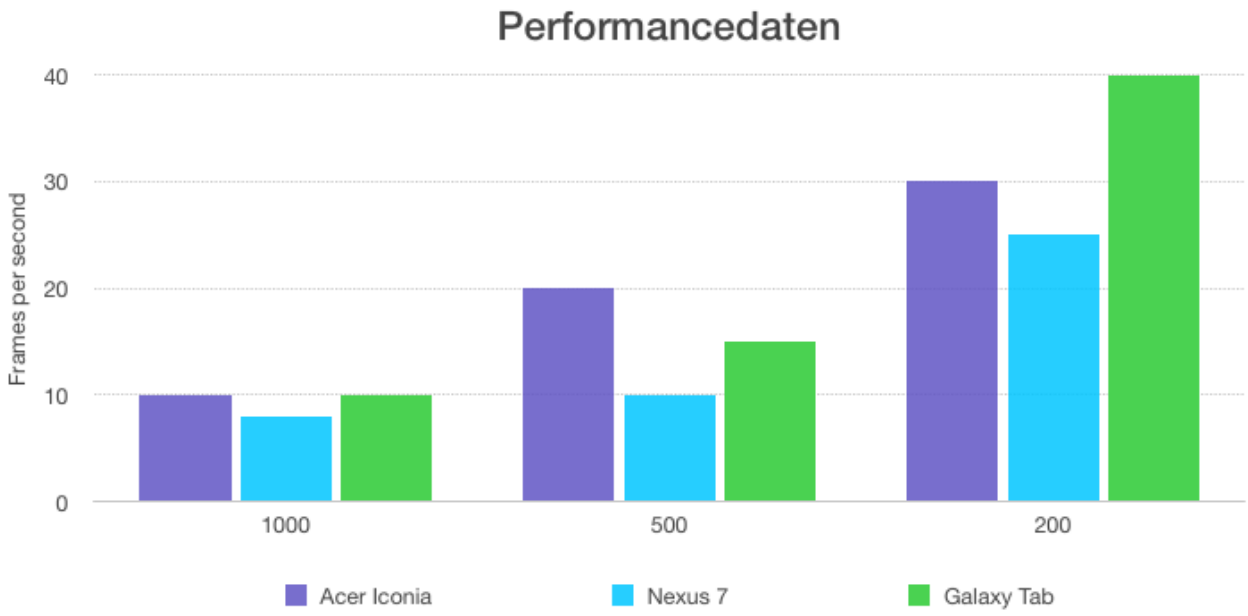
  ⇒ HCP austauschbar

  ⇒ Szene austauschbar



6. Software: Mobile Geräte/libgdx

- Szenen
  - Einfacher Würfel
  - Szene mit komplexen 3D-Modell (.obj, .mtl, etc.)
  - Performancetest
  - HCP



6. Software: Algorithmus

Lösung:

1. Modell platzieren ⇒ globale Koordinaten
2. Off-Axis-Perspektive berechnen

2.1 Frustum Drehung

2.2 Frustum auf Kopfposition anpassen

3. Rotation der Szene um Deviceorientierung
4. Rotation der Szene um Kopforientierung

7. Zusammenfassung

- Funktionierende Middleware
  - Wiederverwendbar/Modular
  - TCP/IP, Sockets
  - Config-Datei
- Grundgerüst der libGdx HCP Anwendung
  - Einfache HCP
  - Rotationen werden behandelt
  - Parallelverschiebungen
- Was fehlt:
  - Total Positionsinvarianz
  - Performanceverbesserungen

8. Fazit: libgdx

- Pro:
  - Deployment auf viele Plattformen möglich
  - Architektur des Frameworks übersichtlich
- Con:
  - Nicht ausgereifte 3D API
  - “vertraute” Methoden nicht nativ verfügbar
    - glFrustum()
    - gluLookAt()
    - …

⇒ Unity3D besser geeignet?

8. Fazit: allgemein

- Unterschätzte Komplexität des Algorithmus
- Spätes Testen mit Live-Daten
- Zeitplan suboptimal

- libgdx Anwendung mit Grund-HCP
- Deployment auf versch. Plattformen
- Stabile Middleware

Fragen?

Vielen Dank für die Aufmerksamkeit!

Quellen

[1] Kooima, R., Generalized Perspective Projection, 2008

[2] Francone, J. Using hte User's Point of View for Interaction on Mobile Devices, IHM '11, 2011

[A] http://1.bp.blogspot.com/-YtjbFWKAu1Y/UXchpyaSkCI/AAAAAAAAA7I/OJjp89MXYy0/s1600/HCpersp520wide.jpg

[B] http://cdn.slashgear.com/wp-content/uploads/2009/03/lenovo_thinkcentre_a58.jpg