

Fakultät Informatik

Institut für Software- und Multimediatechnik

Juniorprofessur Software Engineering Ubiquitärer Systeme

SEUS-Komplexpraktikum

Präsentation: Abschlussbericht

Dresden, 19. Juli 2013



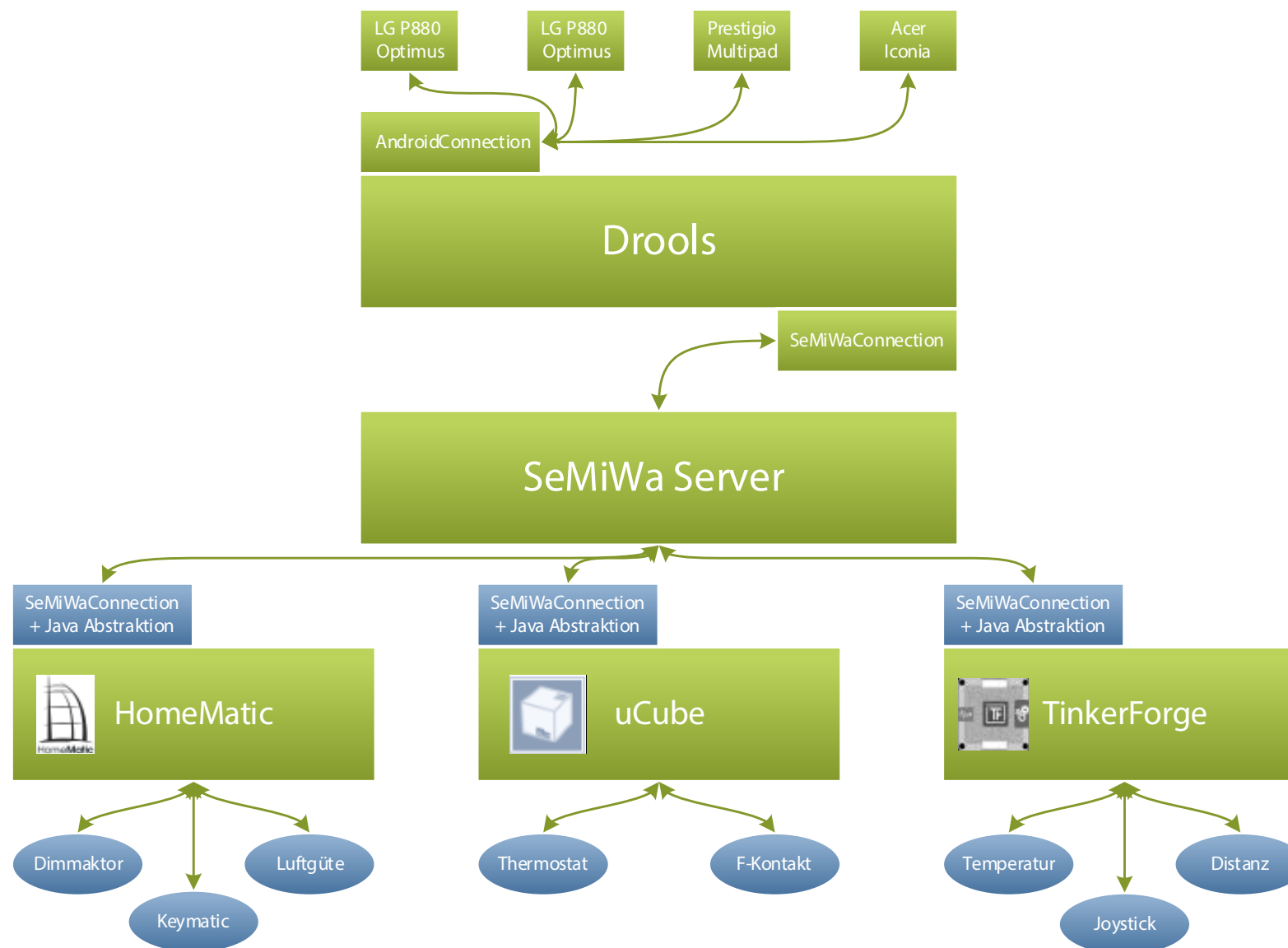
Problemstellung

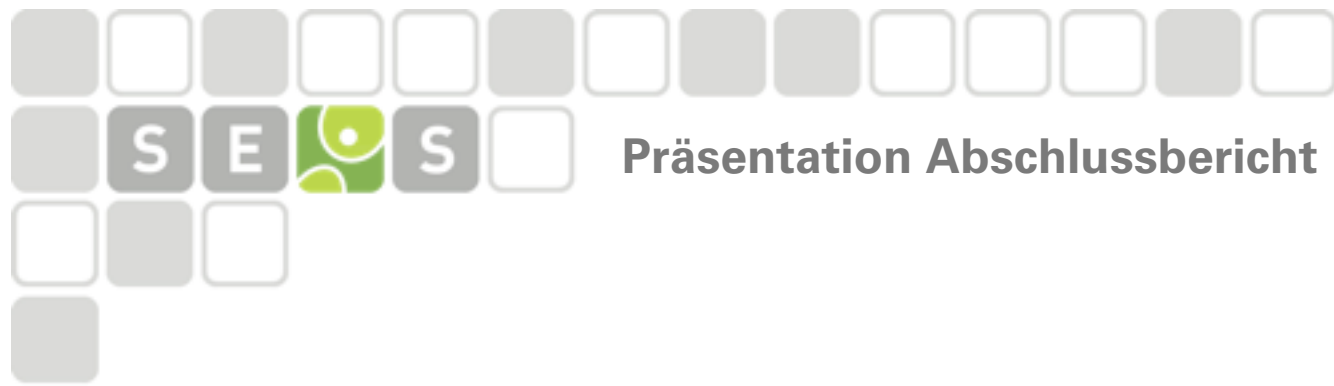
- zunehmende Anzahl von Herstellern
- Hardware und Software zu erschwinglichen Preisen
- proprietäre Lösungen – schlechte Integration/Kombinierbarkeit
- Schnittstellen häufig vorhanden



**MELISSA – modellbasierte Integration von Smart Home
Sensorik und Aktorik**

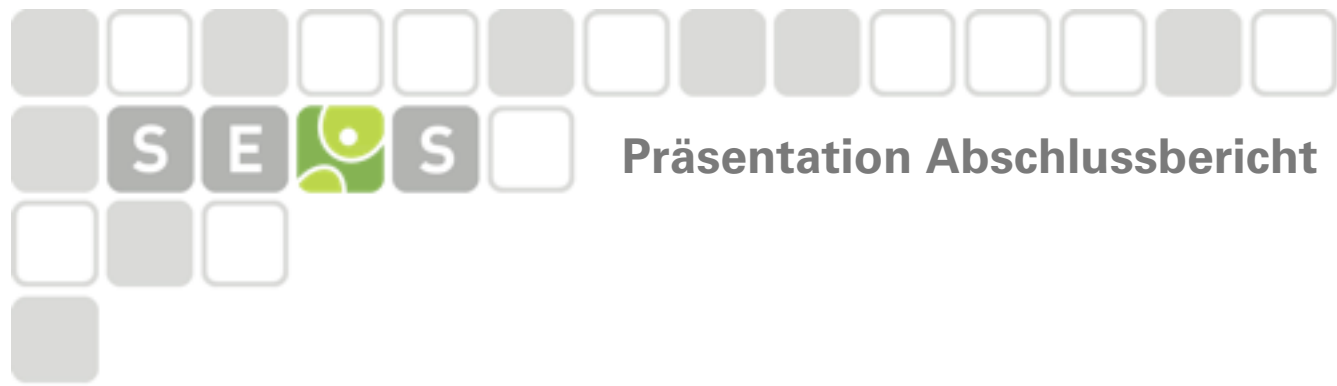
Abstraktion





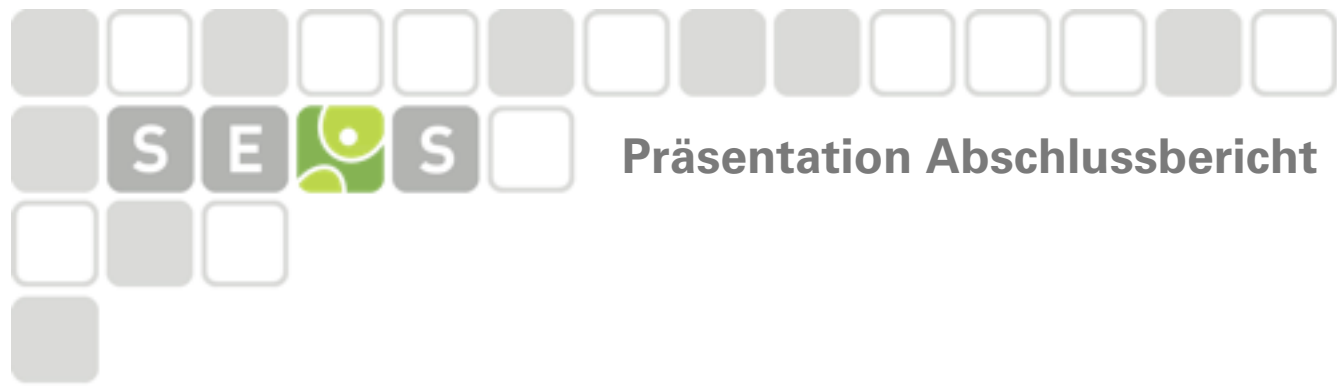
Aufgabenstellung und Aufgabenaufteilung

- Abstraktion der drei SmartHome-Systeme: Bereitstellung einer gemeinsamen Schnittstelle und beidseitige Steuerung von Aktoren/Sensoren
 - Tinkerforge (gemeinsam)
 - HomeMatic (Philipp)
 - MAX! (Anika, Nico, Stephan)
- Protokollierung der Treffen (Stephan)
- Hinzufügen von weiteren Instanzen in das SeMiWa-Modell (Anika)
- Ableiten eines geeigneten Vorgehensmodells zur Erweiterung der Abstraktion um Sensoren (gemeinsam)



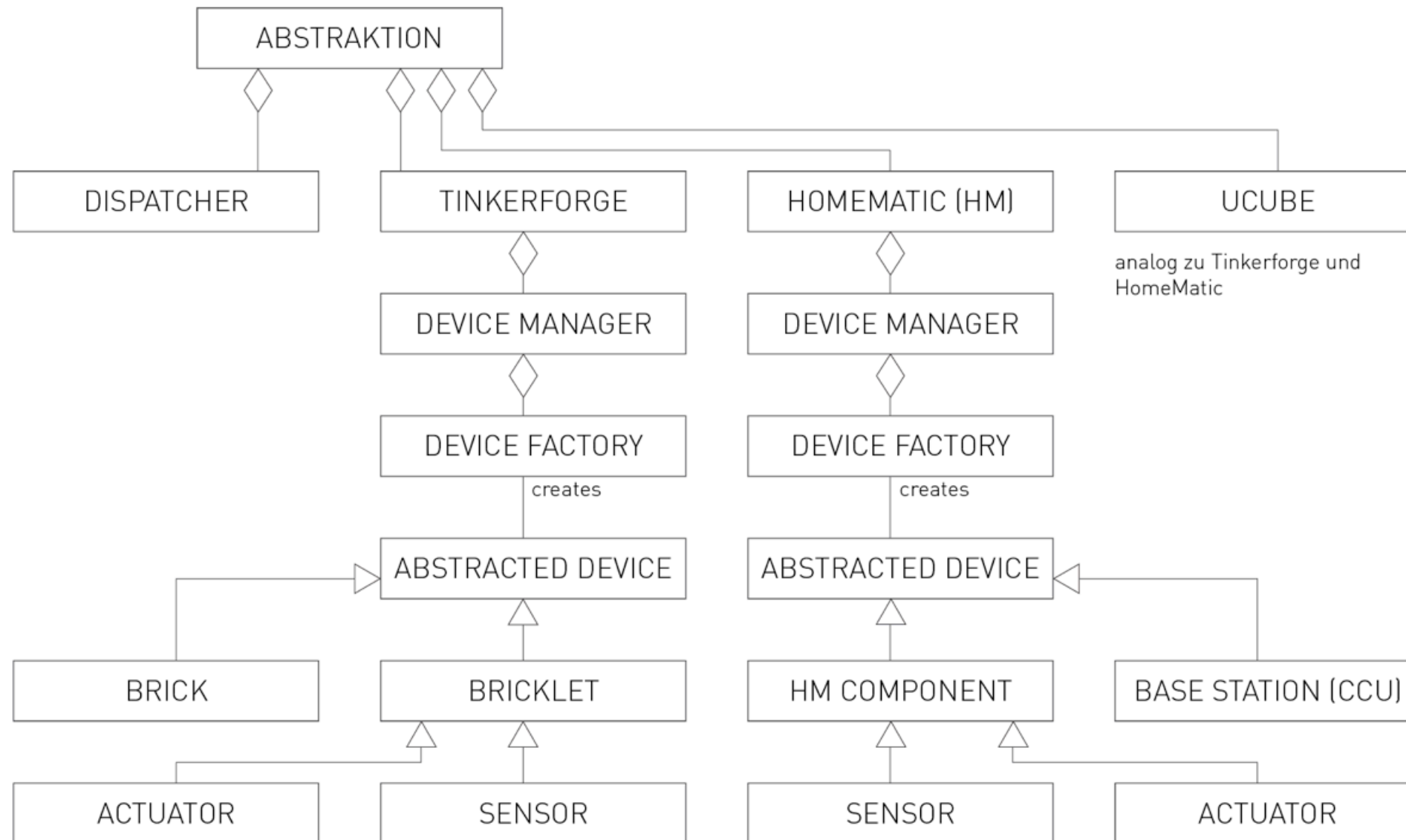
Organisation

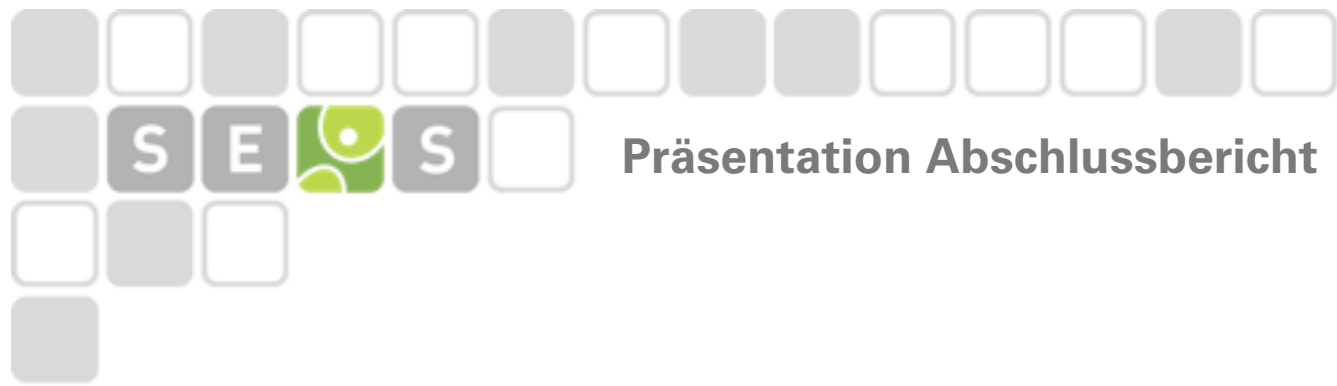
- Treffen 2x pro Woche im Falkenbrunnen
- Co-Programming
- individuelle Arbeit an den Implementierungen der Abstraktionen
- Protokolle im Wiki
- SVN



Verfahrensmodell

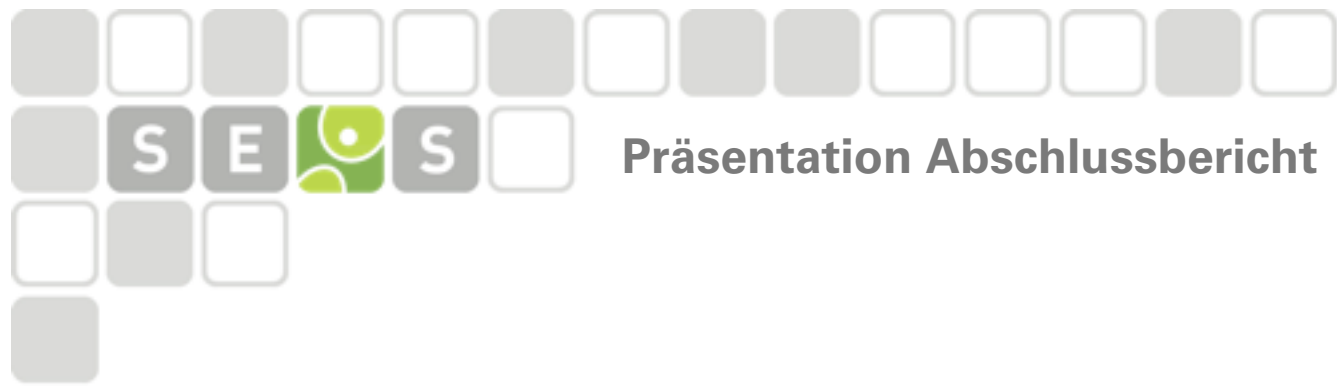
- API-Anbindung (System-spezifisch)
- Integration neuer Sensoren/Aktoren
 - Erweiterung des semantischen Modells um zugehörige Instanzen
 - Übermitteln des Datasheets (Registrierung bei SeMiWa)
 - uid, deviceName, deviceUnit, deviceType, class
 - Erweiterung der DeviceFactory um neues Gerät
 - Generierung der Klassen
 - Geräte- und Kommandoklassen





Prototyp

- abstrahiert (fast) alle am Lehrstuhl vorhandenen Sensoren und Aktoren
- 28 Geräteklassen
- Dispatcher zur Kommunikation mit SeMiWa
- mit freundlicher Vorarbeit von Martin bei der TinkerForge-Abstraktion 😊
- semantische Beschreibung aller Geräte

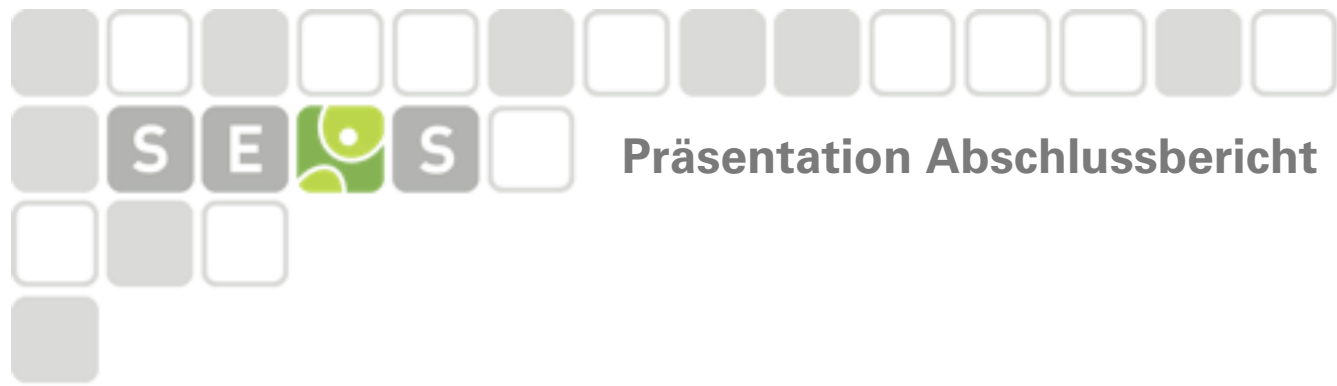


Verwendete Technologien/Bibliotheken

- JAVA SE 6
- SeMiWa (Connection und Node)
- GSON 2.2.2
- Java-WebSocket
- Apache XML-RPC (3.1.2)

Probleme

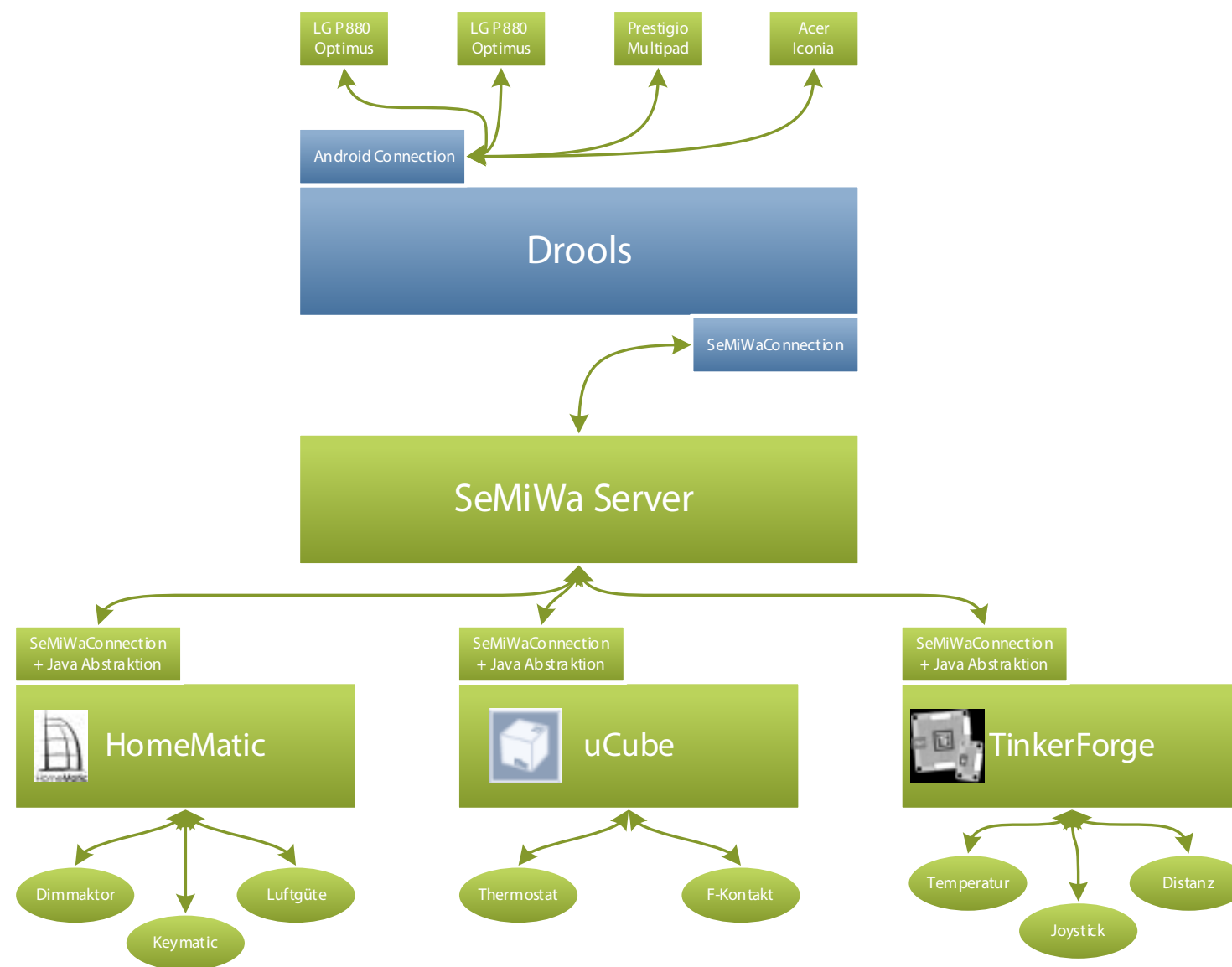
- HomeMatic
 - Abstraktion benötigt (lokalen) XML-RPC-Server, um eingehende Nachrichten der HomeMatic-API entgegen zu nehmen
 - Montage des Heizungsstellantriebs und der KeyMatic
 - schwache Dokumentation
- MAX!
 - kein Auslesen von einzelnen Gerätedaten möglich
 - Zuordnung der Daten zu Geräten
 - sehr lange Aktualisierungs- bzw. Antwortzeiten



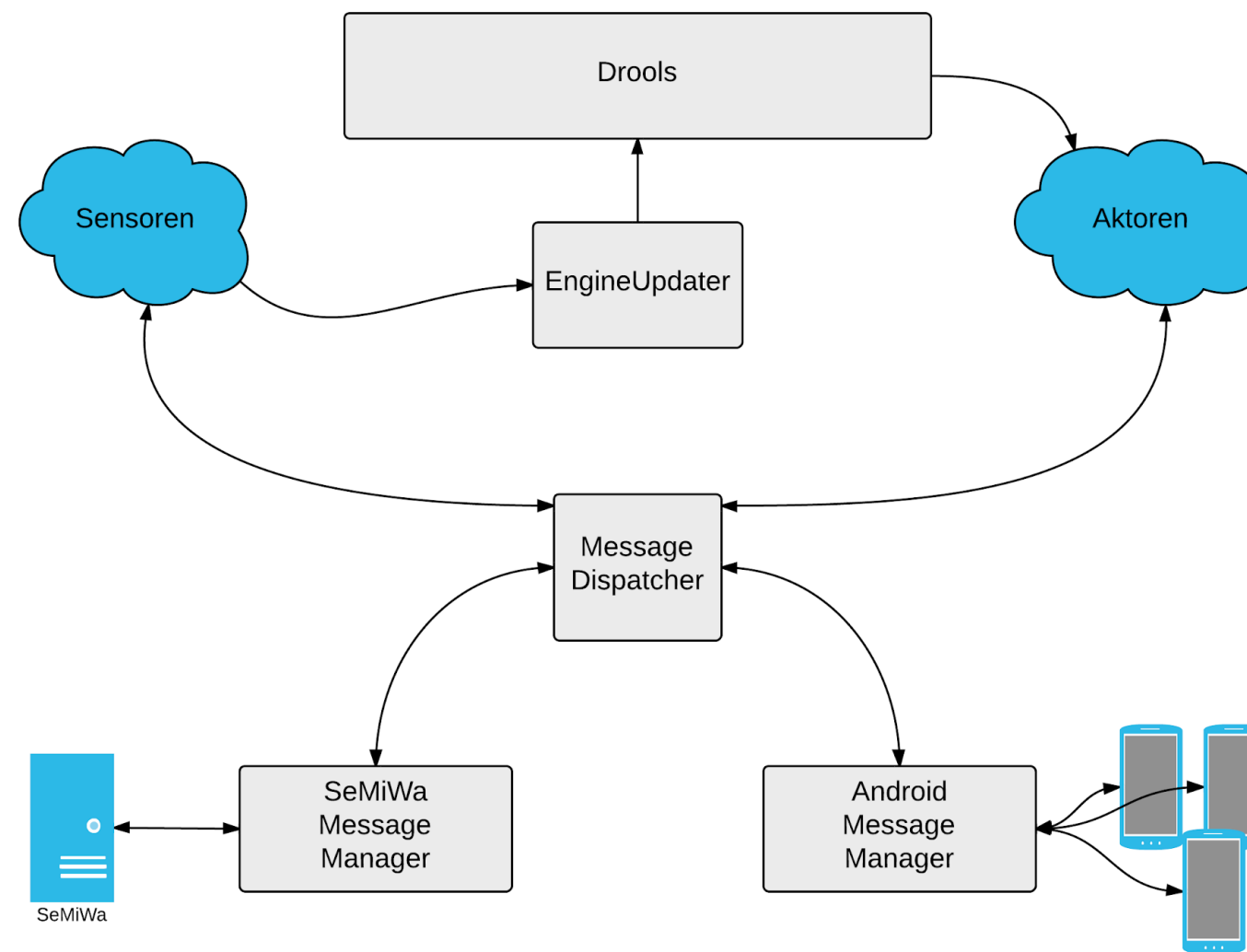
Zeitplan nach der Zwischenpräsentation

- Implementierung der Abstraktionen für HomeMatic und MAX! (6 Wochen)
- Erweiterung des semantische Modells + Vorgehensmodell (1 Woche)
- Testen (1 Woche)
- Abschlussbericht und Abschlusspräsentation (1 Woche)

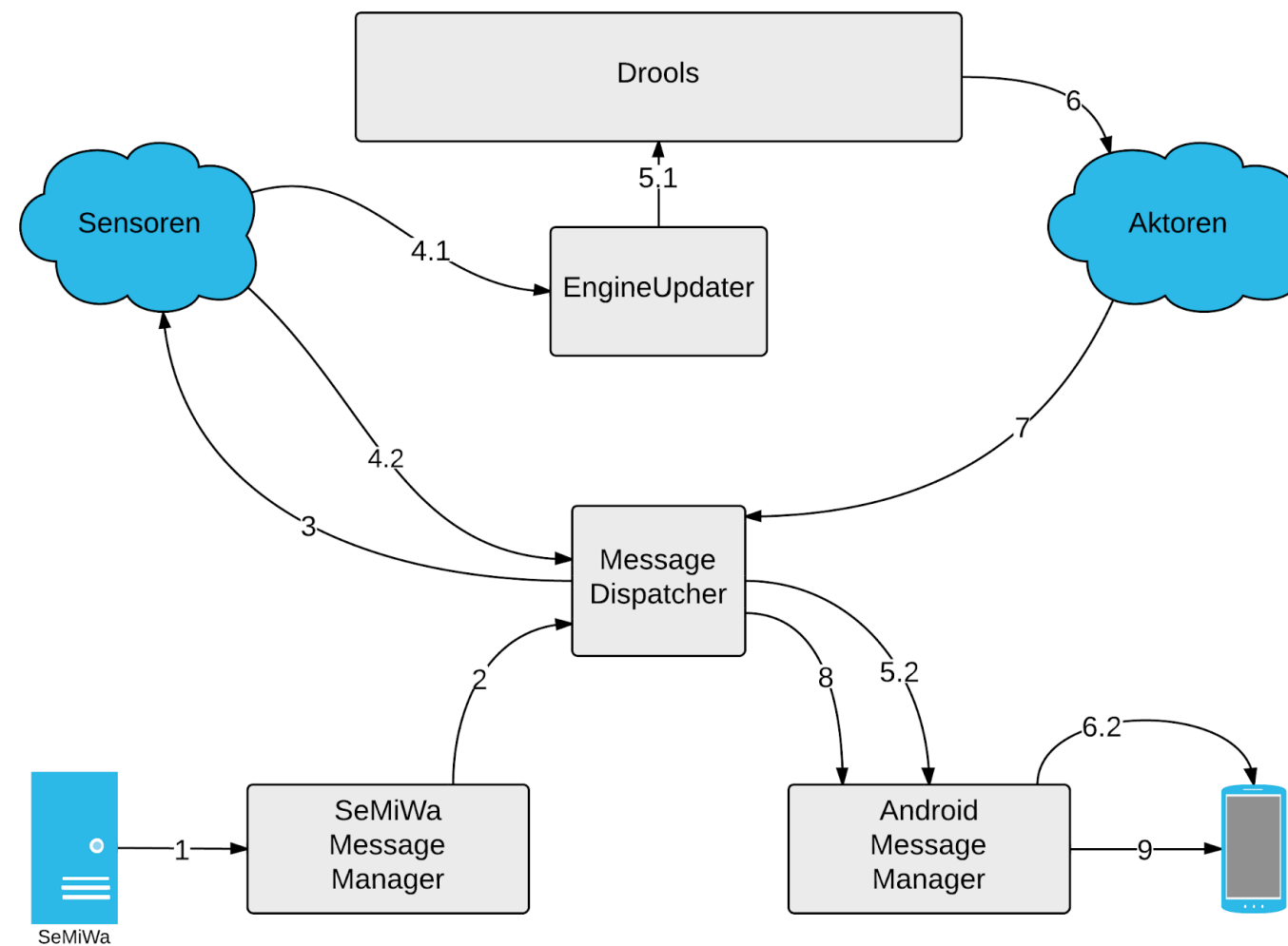
Drools



Architektur

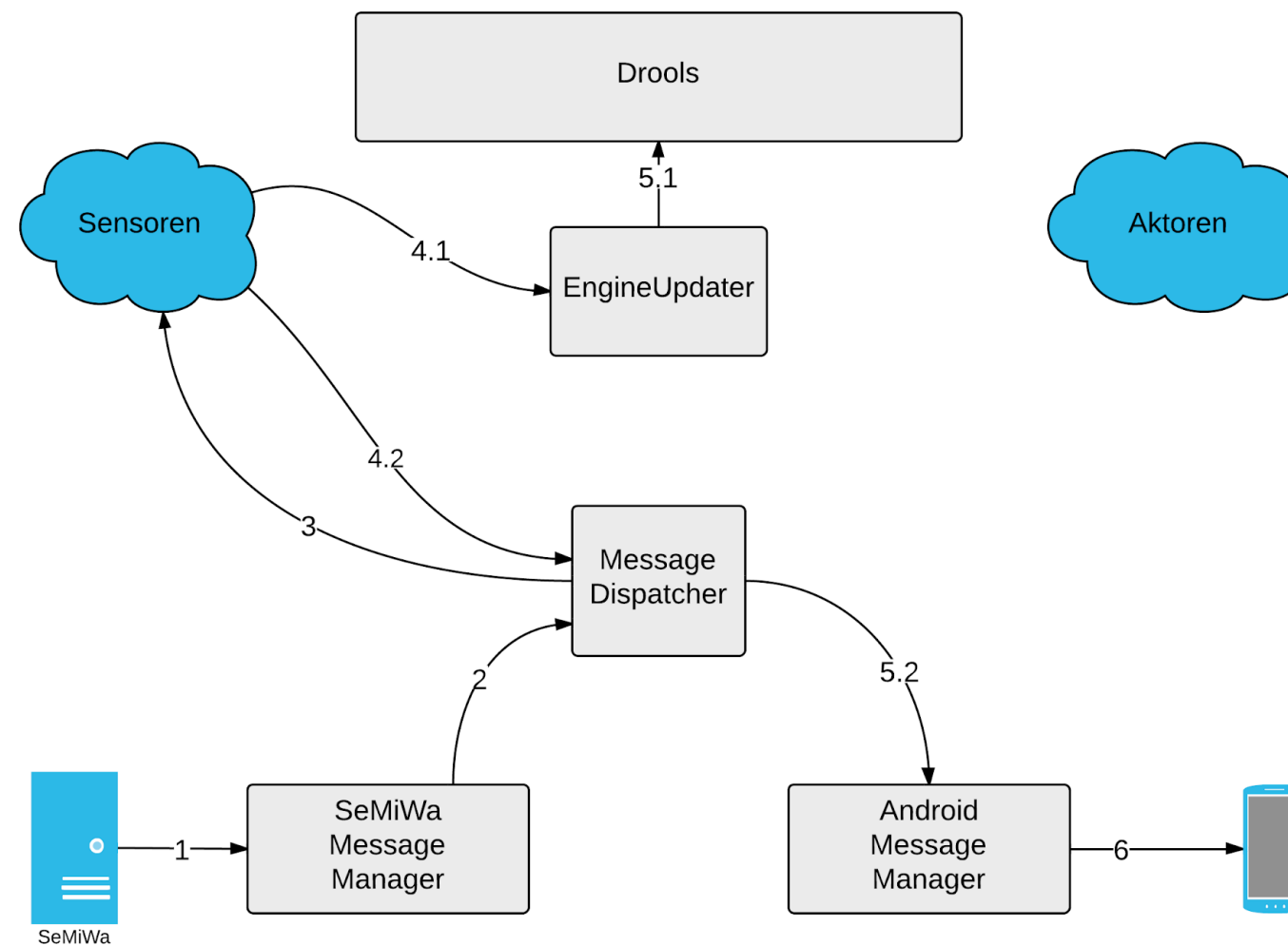


Diagramme



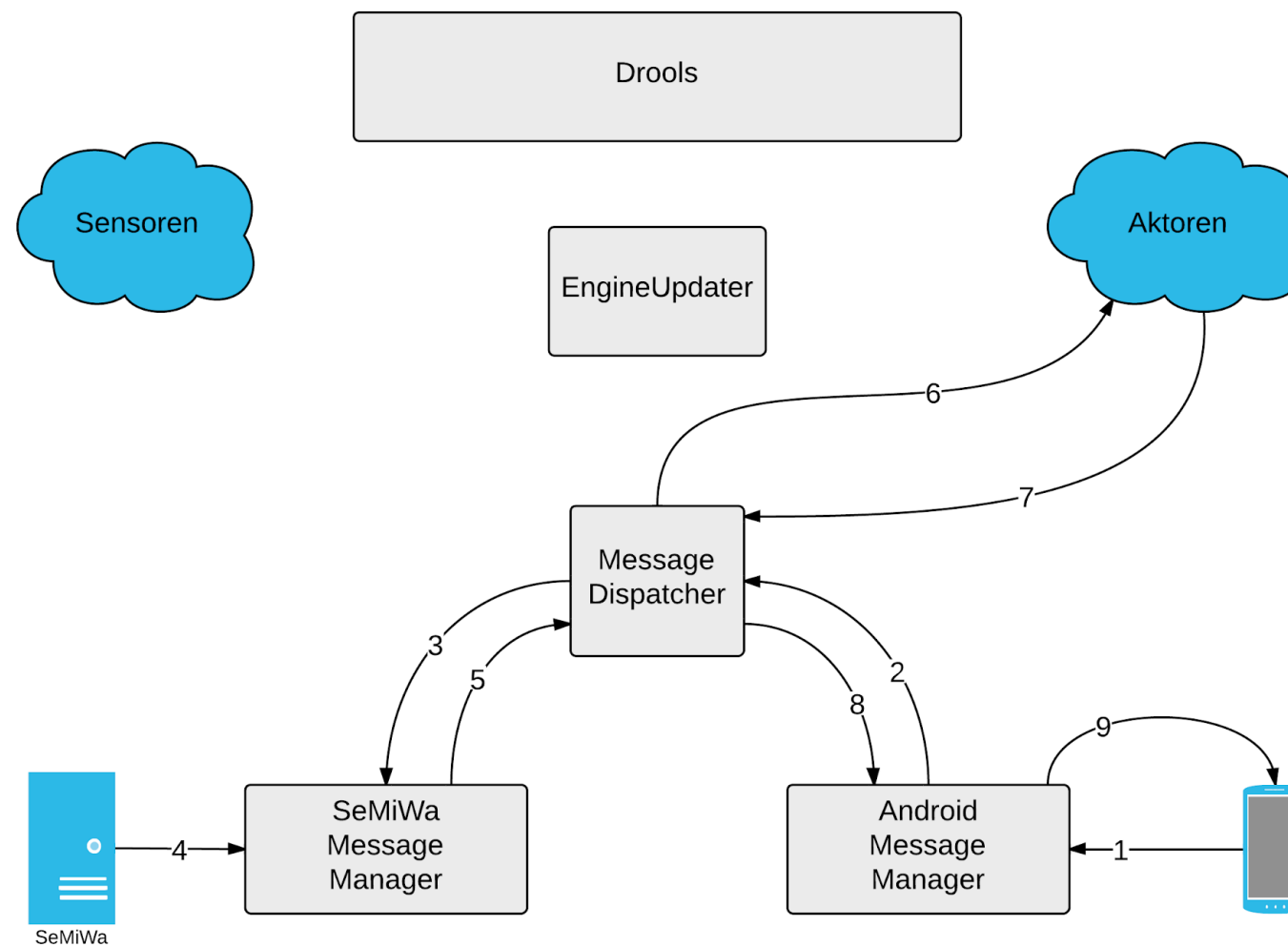
Sensorupdate von SeMiWa mit Akteurupdate

Diagramme



Sensorupdate von SeMiWa ohne Akteurupdate

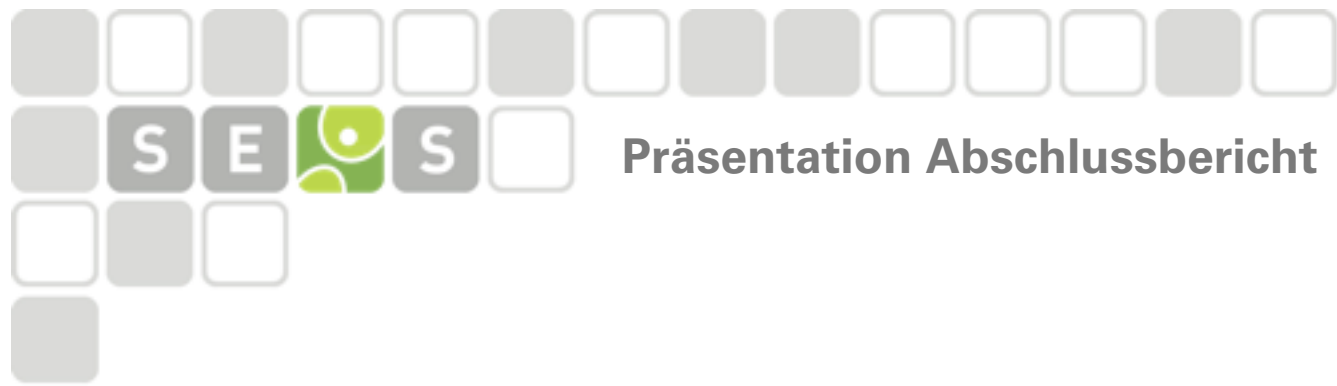
Diagramme



Aktorupdate nach Befehl von Drools

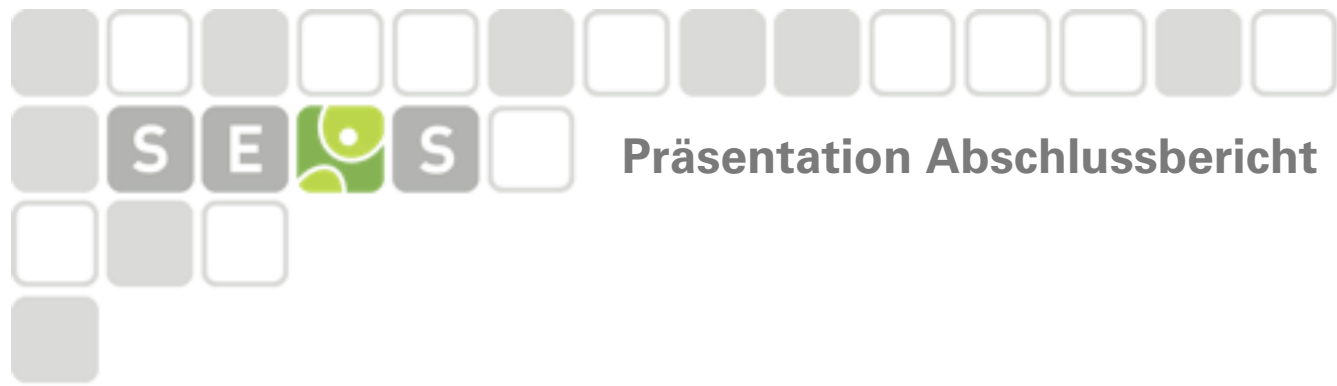
Prototypen

- Dummy-Generator für SeMiWa-Events
 - minimales Interface zum Testen ohne SeMiWa
- Drools Engine
 - Regeln in *.drl-Dateien (eine Datei = eine Regel)
 - Hinzufügen von Regeln **nicht** während des Betriebs
 - Drools Guvnor könnte dies lösen
 - Komponenten werden automatisch hinzugefügt
 - Android-Geräte melden sich automatisch an
 - Durchschnitt mittels History
 - Sensoren und Aktoren generisch



Prototypen

- Mögliche Erweiterung:
 - WebInterface
 - würde property-Dateien erübrigen
 - Regelerstellung mit Hinzufügen könnte hier stattfinden
 - Drools Guvnor
 - gruppenübergreifend einheitliches Schema der Komponentenrepräsentation

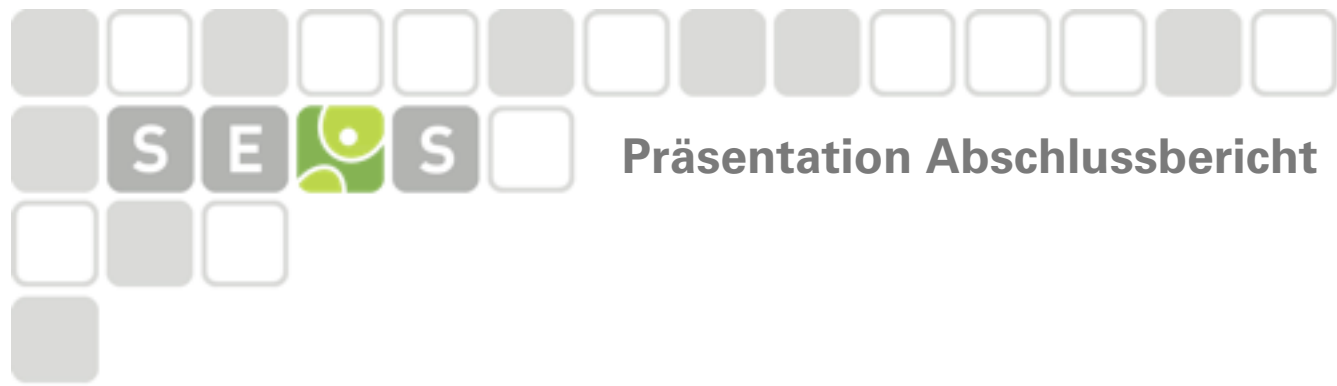


Verwendete Technologien

- Maven zur Abhängigkeits- und Projektverwaltung
- Eclipse
- SVN und Wiki
- Java 1.7
- Kommunikation zu Android: WAMP
- SeMiWa
- Drools

Gruppenaufteilung

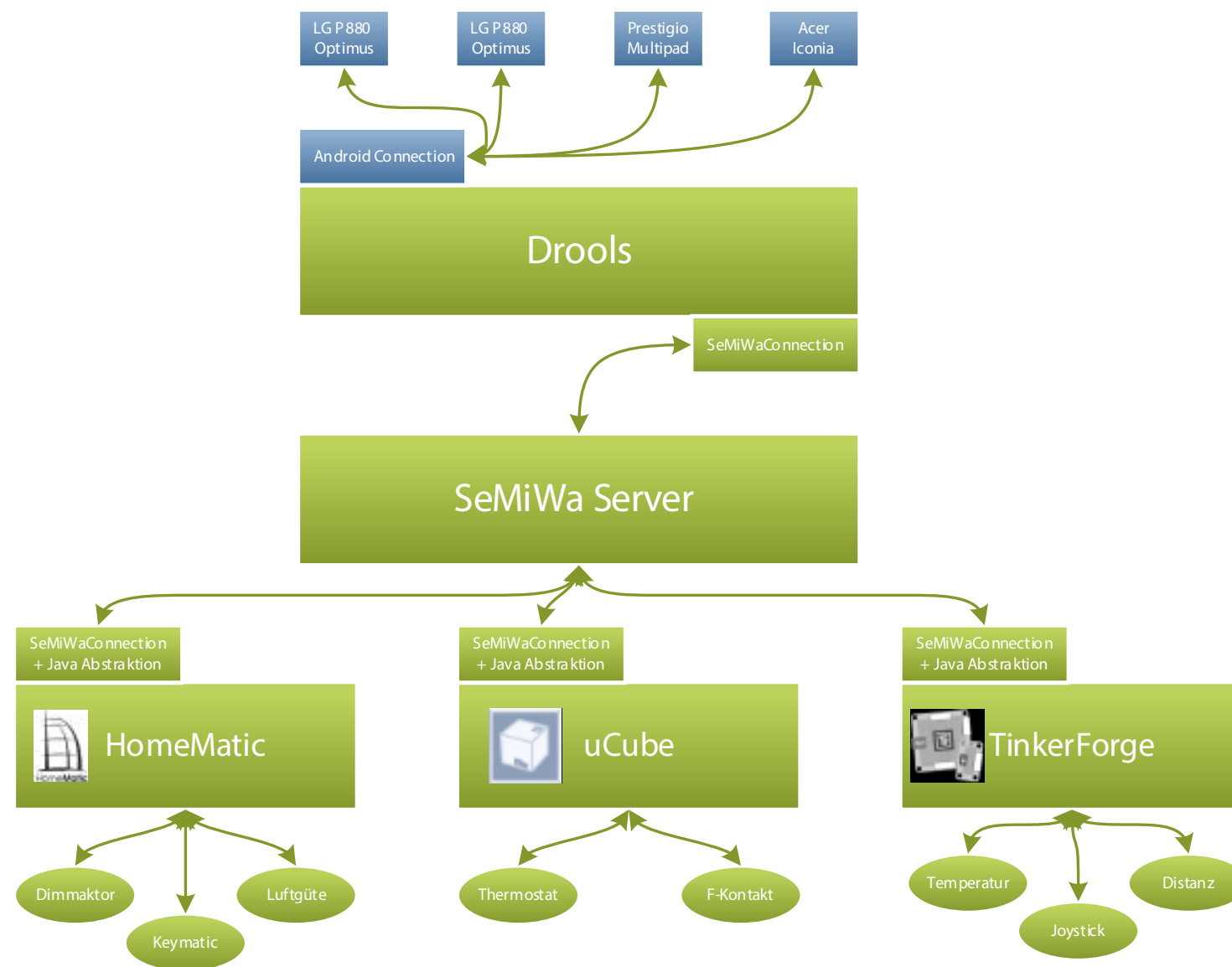
- Stefan
 - Netzwerk-Kommunikation zu SeMiWa und Android
- Florian
 - Drools Engine
 - Message Management
- Jonas und Torsten
 - Drools-Regeln
 - Komponentenabstraktion



Aufgetretene Probleme

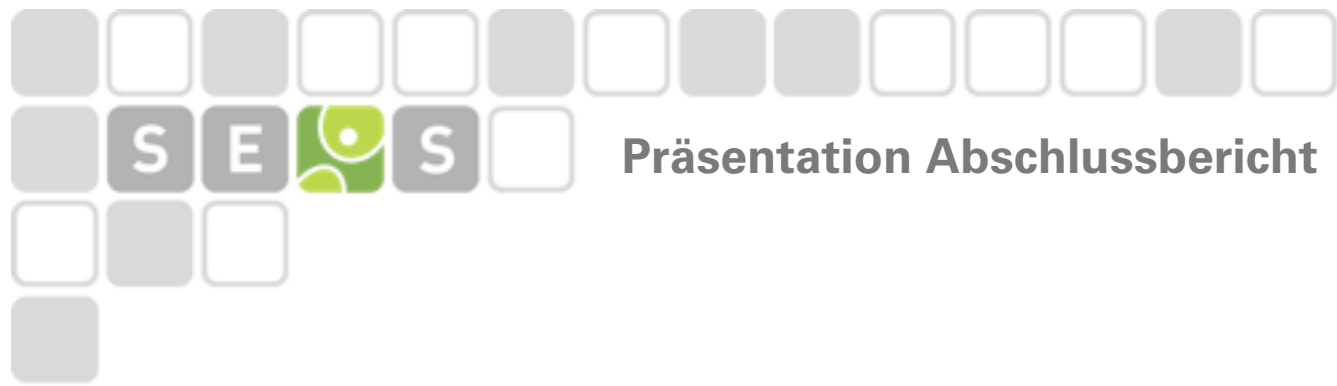
- Kommunikation
 - Drools – Android
 - SeMiWa – Drools
- Drools-Setup bzw. Update
- Anbindung der realen Bauteile
- Gruppenübergreifende Kommunikation

Android



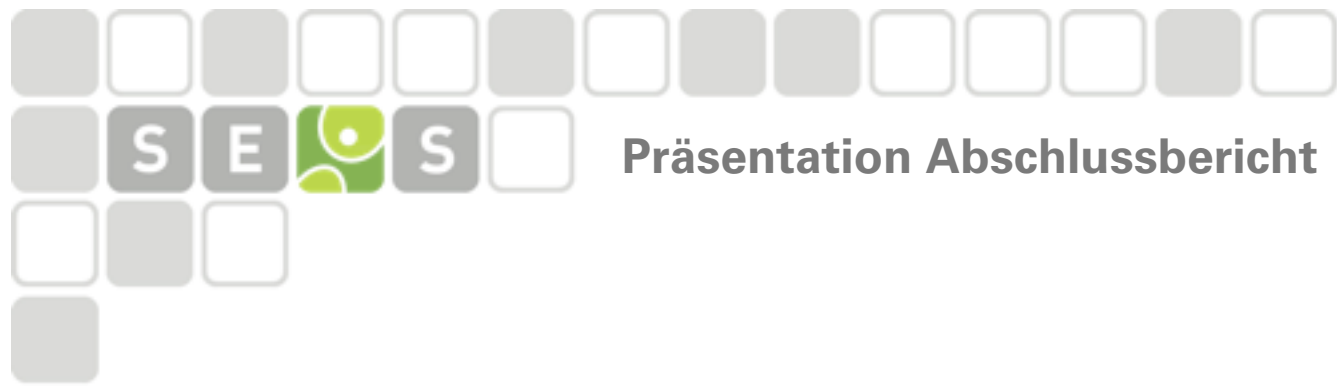
Überblick

- Team
 - Peter Heisig – Projektleitung & Kommunikation
 - Stefan Vogt – stellv. Projektleitung & UI-Design
 - Maximilian Hartig – Spezifikation & Controlling
 - Tobias Hafermalz – Prototyping & Testing
- Problemstellung
 - Wiederverwendbare Visualisierungskomponenten
 - sinnvolle Aktorsteuerung
 - standardisierte Kommunikation mit der Prozessengine Drools

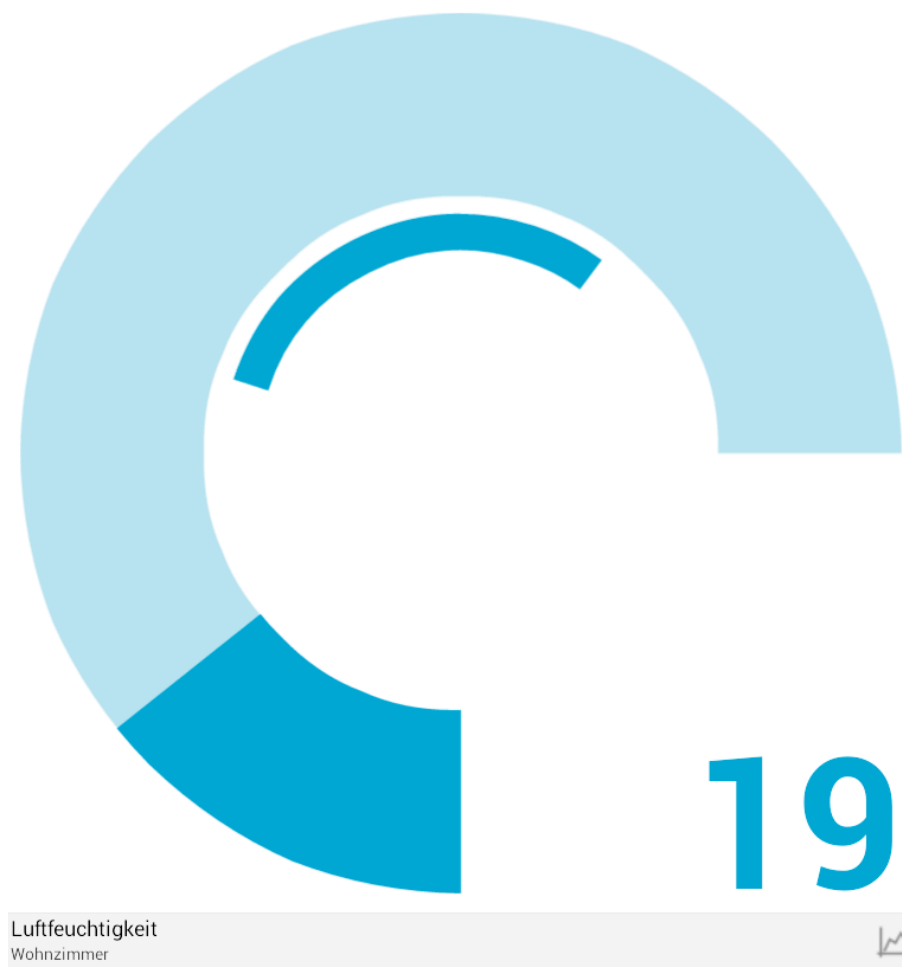


Visualisierungen - Sensorik

- Kontinuierliche Darstellungen
 - Temperatur
 - Luftfeuchtigkeit
 - Umgebungslicht
 - Abstand
- Diskrete Darstellungen
 - Bewegung
 - Tür
 - Fenster
- Trenddarstellungen



Kontinuierliche Darstellungen – Luftfeuchtigkeit



Luftfeuchtigkeit
Wohnzimmer

19

Idealbereich

Wertebereich (100%)

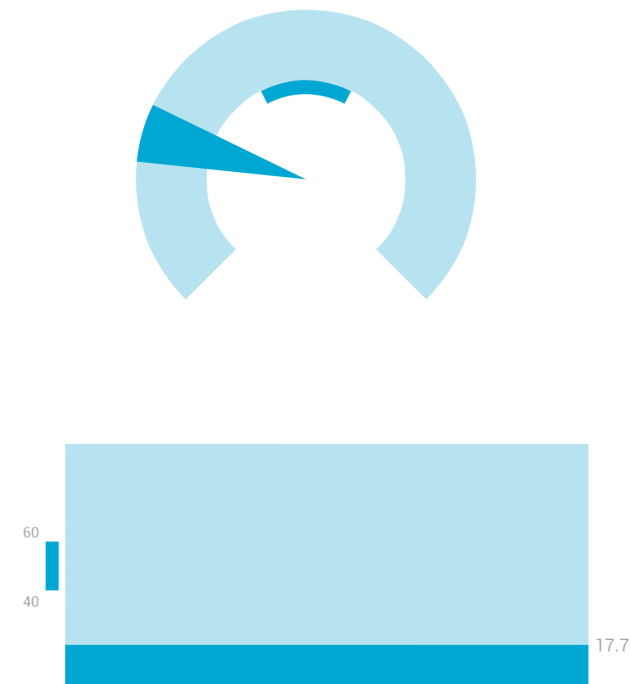
Aktueller Wert (19%)

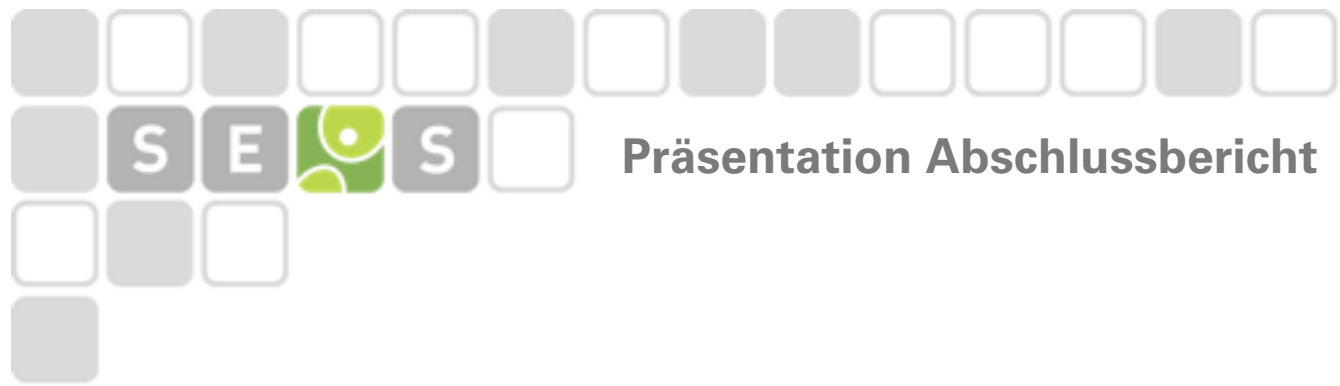
Numerischer Wert

Statusleiste

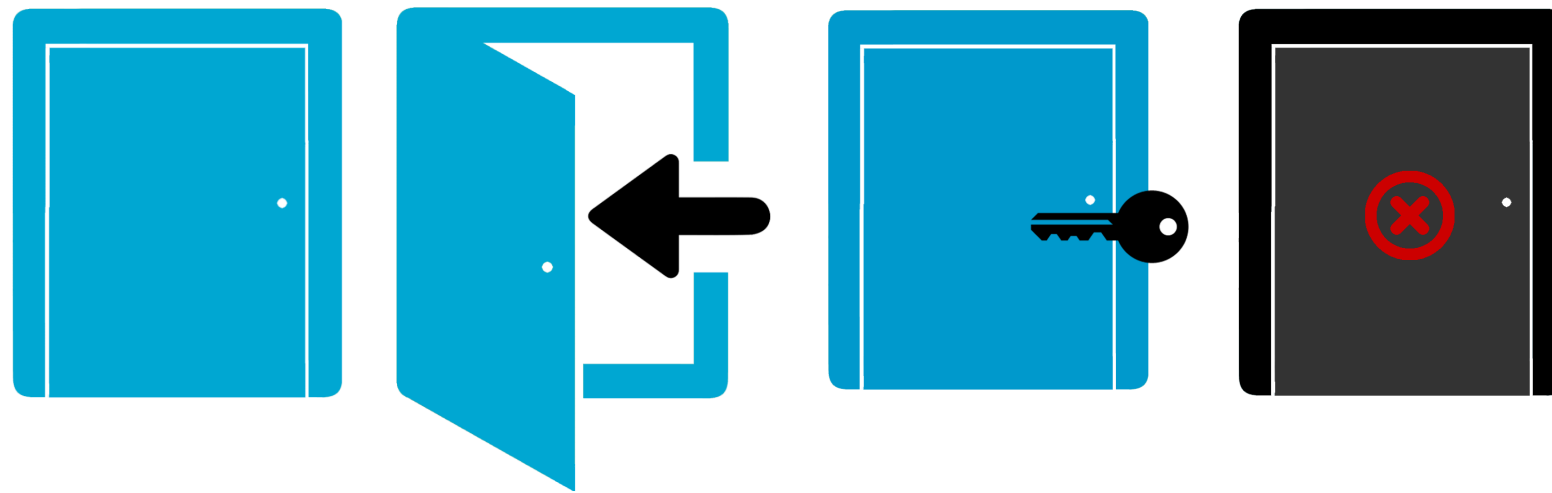
A donut chart with a light blue outer ring and a darker blue inner ring. The inner ring is partially filled, representing 19% of the total. The number '19' is displayed in a large, bold, blue font to the right of the chart.

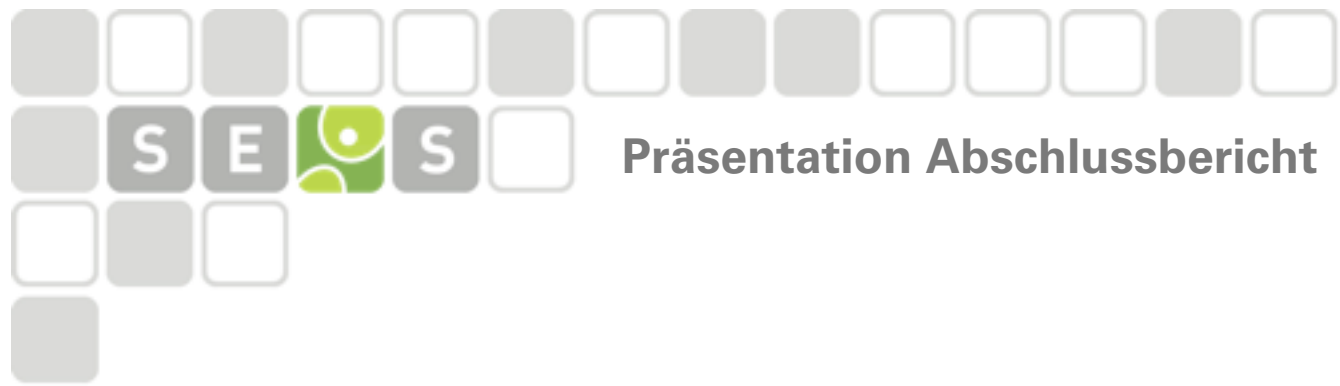
A line graph with a vertical y-axis and a horizontal x-axis. The line starts at the origin (0,0), rises to a peak, falls to a local minimum, rises to a higher peak, and then falls slightly. The overall trend is upward.



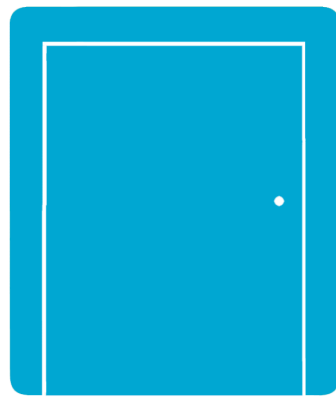


Diskrete Darstellungen – Tür

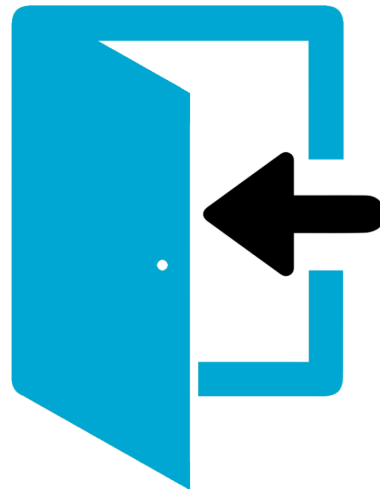




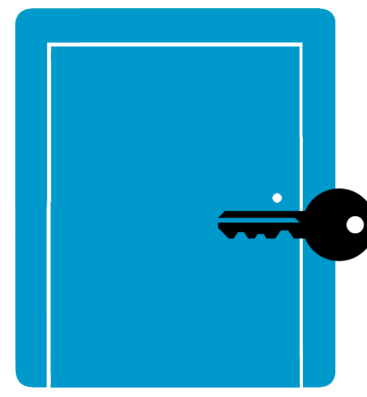
Diskrete Darstellungen – Tür



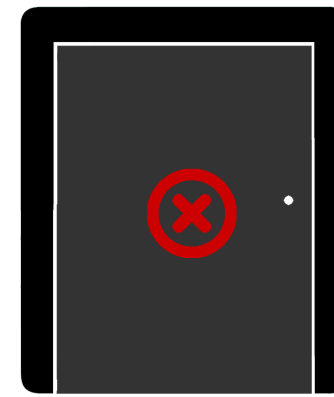
geschlossen



geöffnet

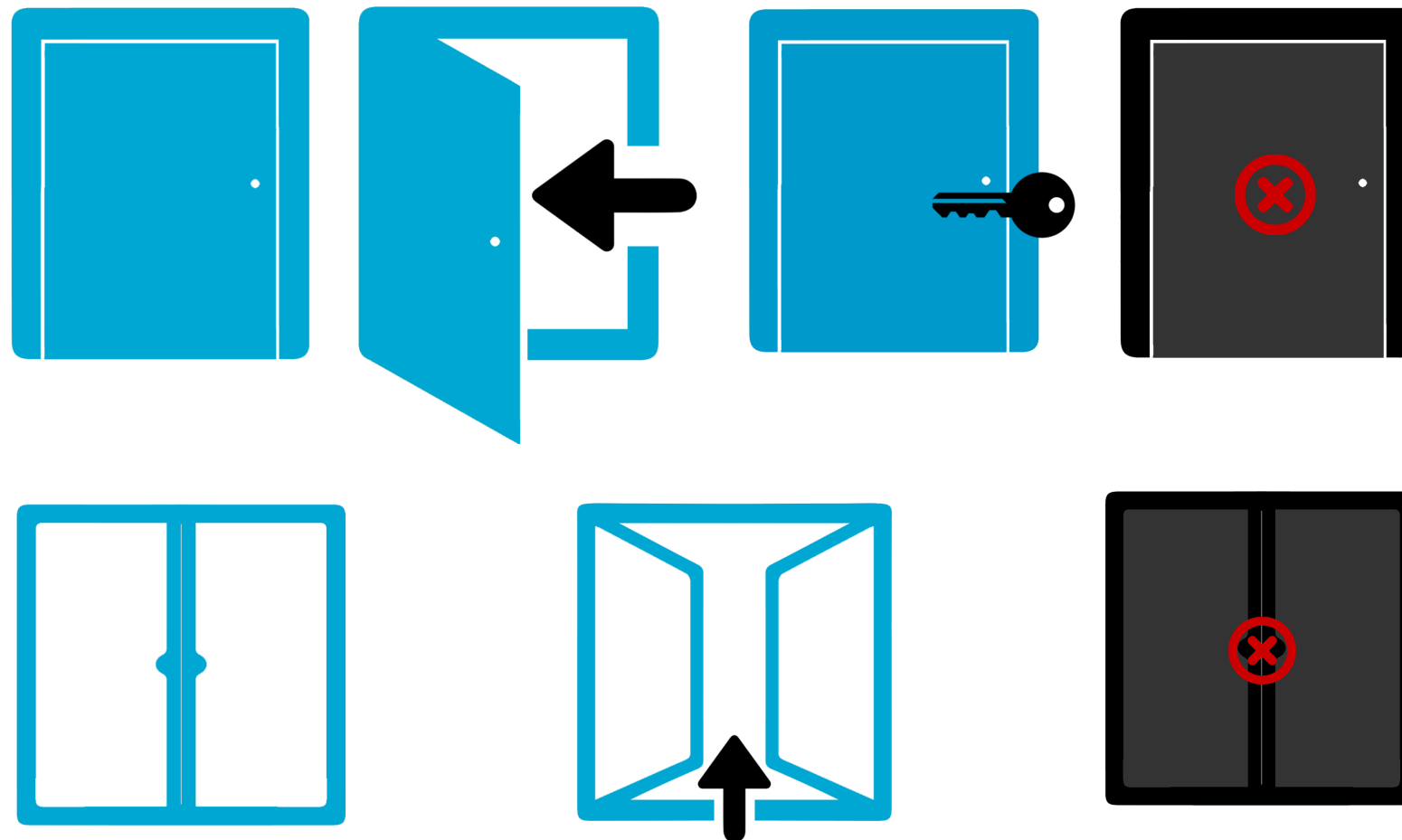


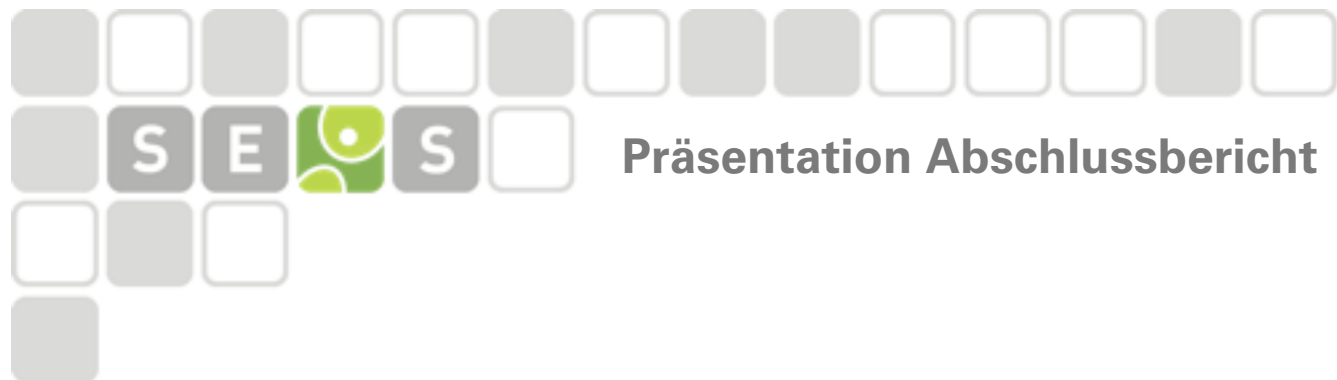
verschlossen



unbekannt

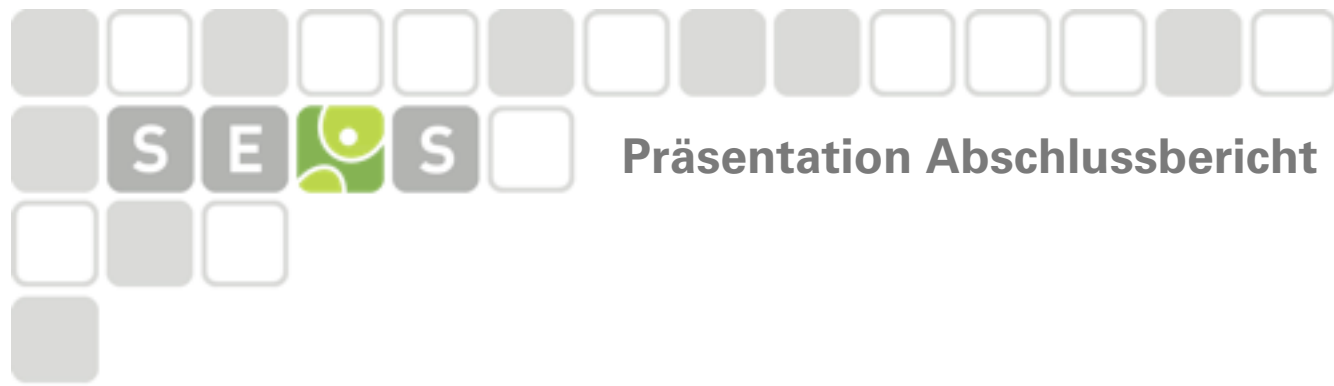
Diskrete Darstellungen – Tür





Trenddarstellungen – Temperatur

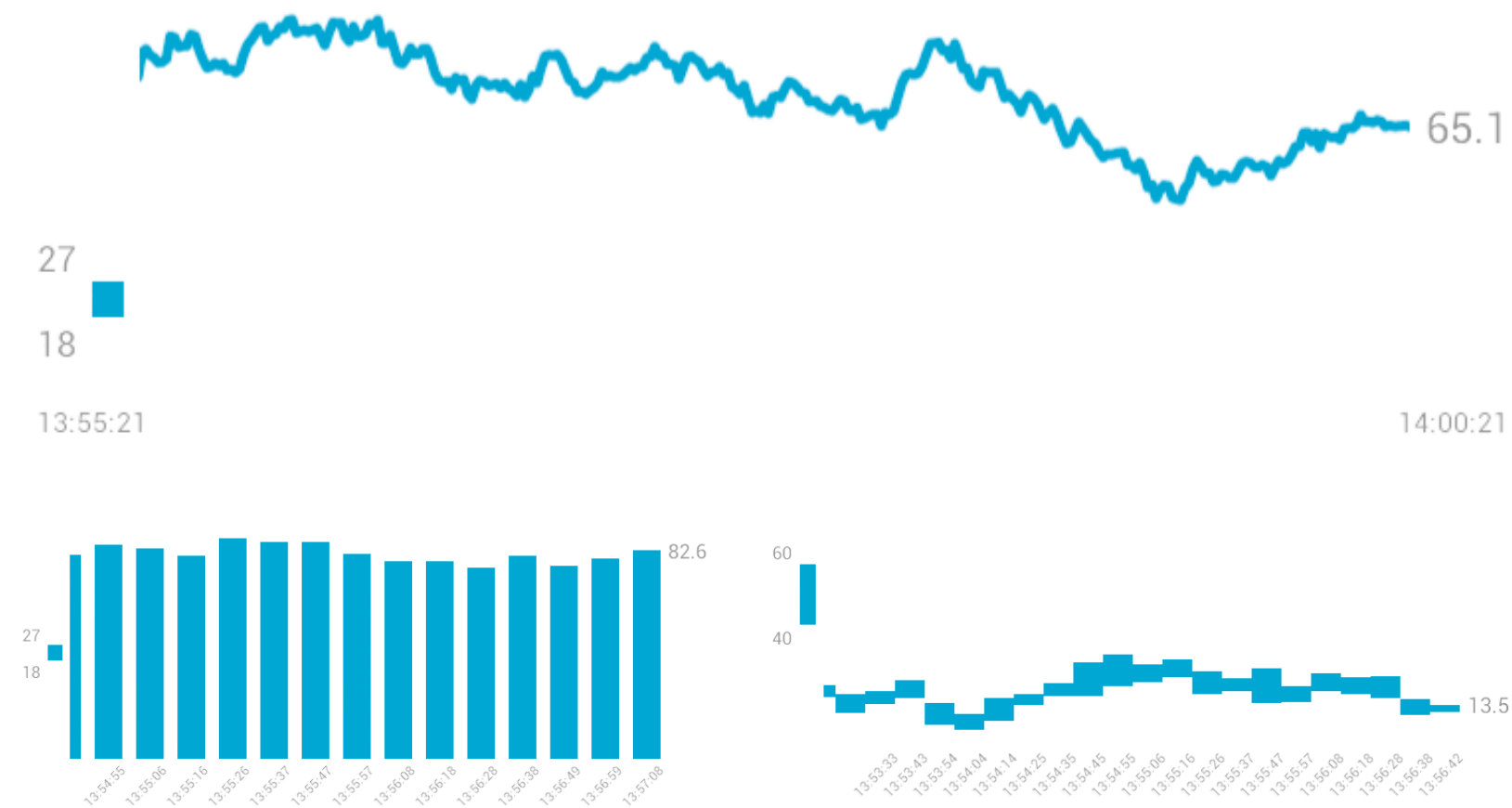


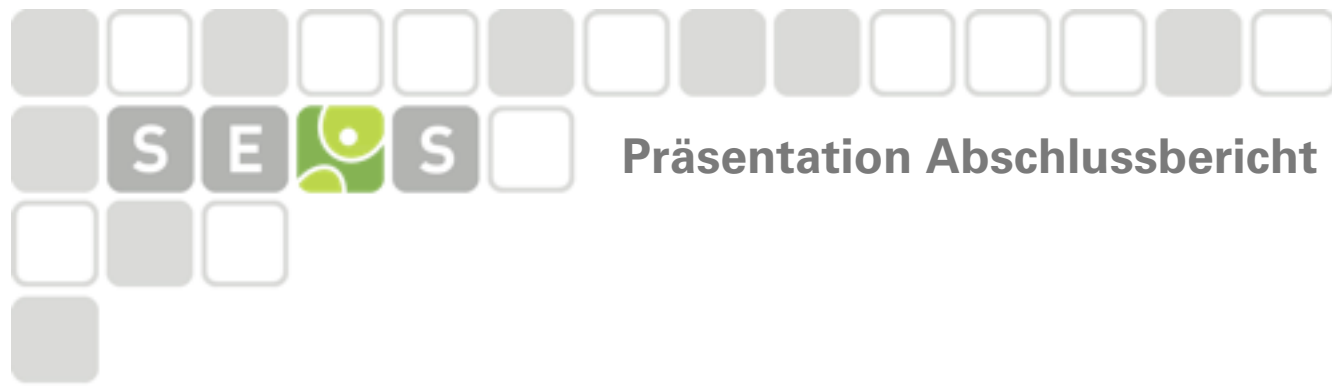


Trenddarstellungen – Temperatur



Trenddarstellungen – Temperatur



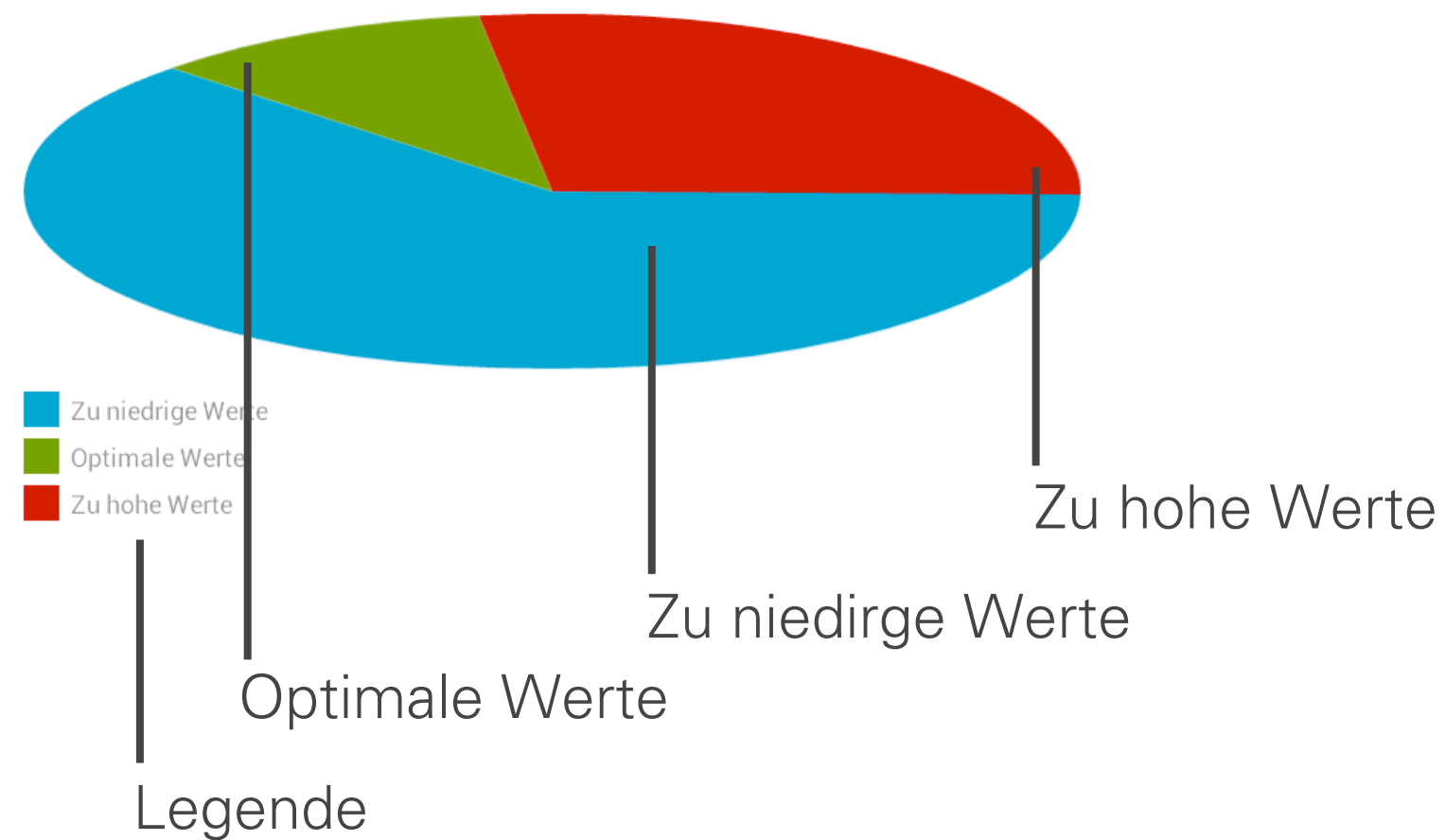


Trenddarstellungen – Luftfeuchtigkeit

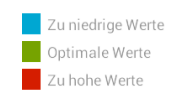
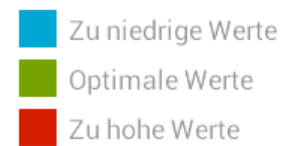


- Zu niedrige Werte
- Optimale Werte
- Zu hohe Werte

Trenddarstellungen – Luftfeuchtigkeit



Trenddarstellungen – Luftfeuchtigkeit



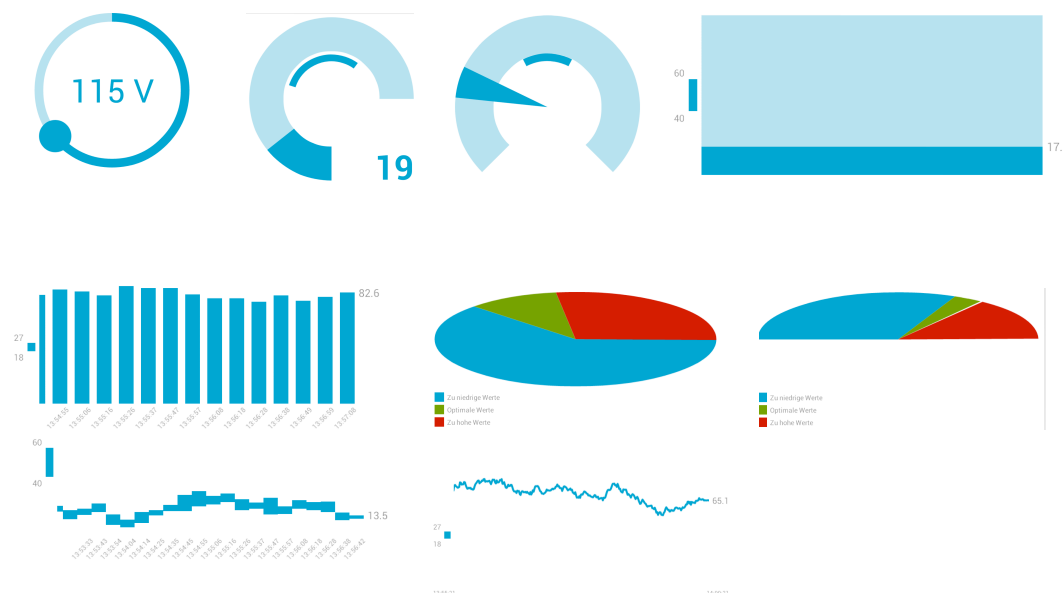
Steuerung – Aktorik

- Diskrete Steuerung
 - Keymatic
- Kontinuierliche Steuerung
 - Dimmer
 - Thermostat

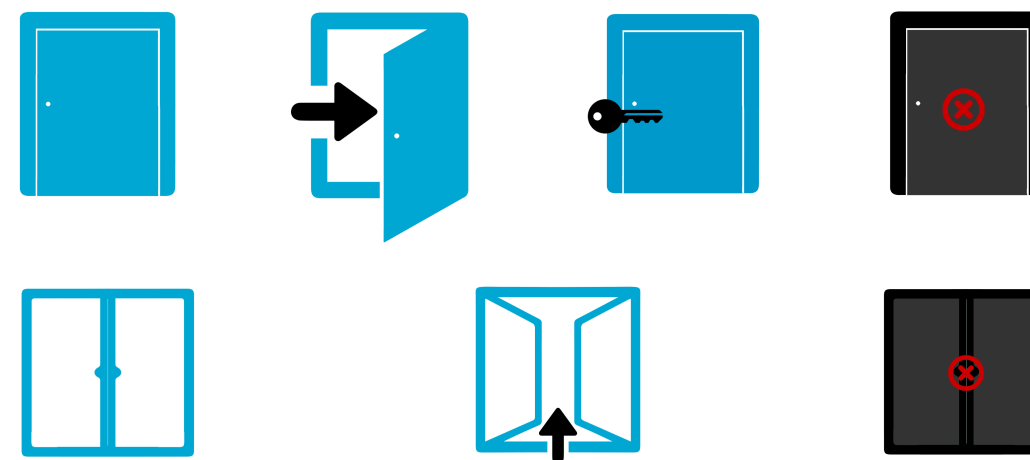


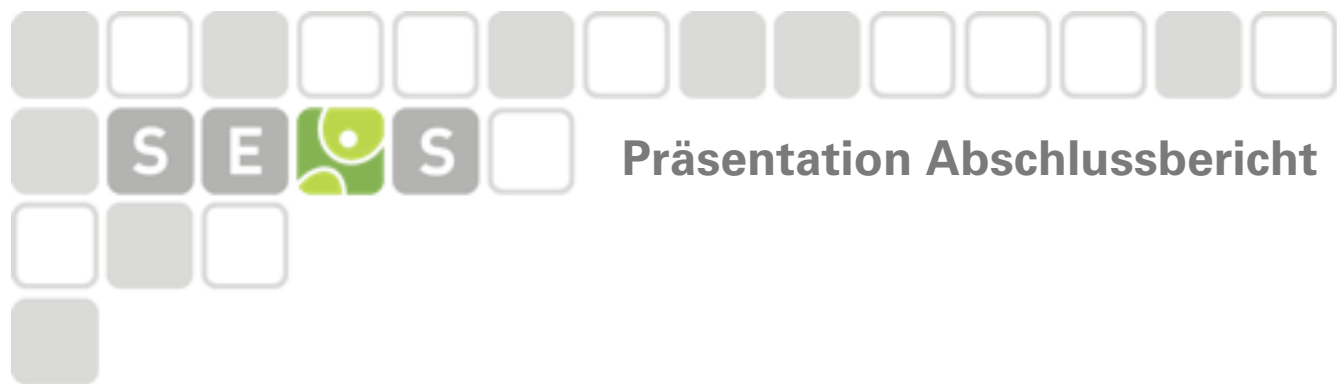
Visualisierungen

- Heizung
- Temperatur
- Luftfeuchtigkeit
- Umgebungslicht
- Abstand



- Tür
- Fenster

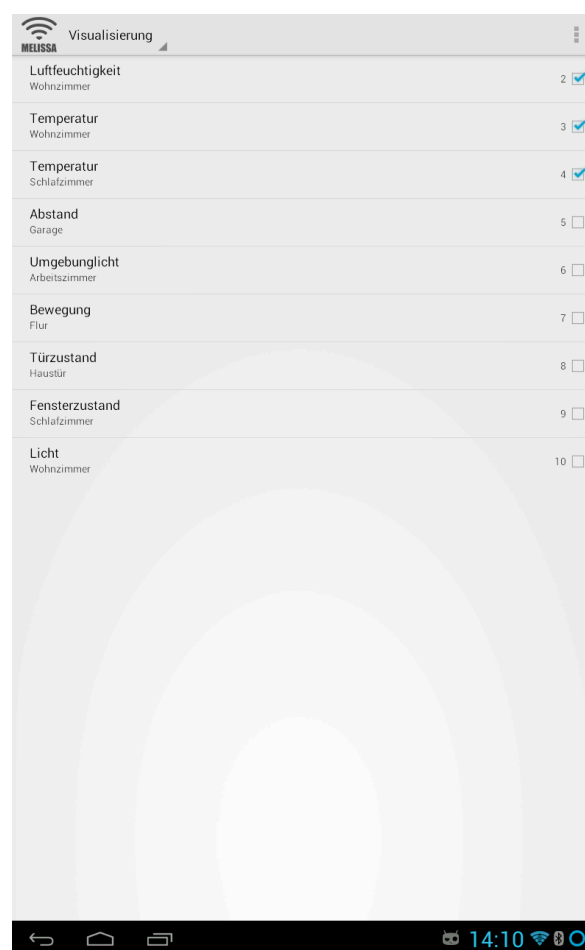




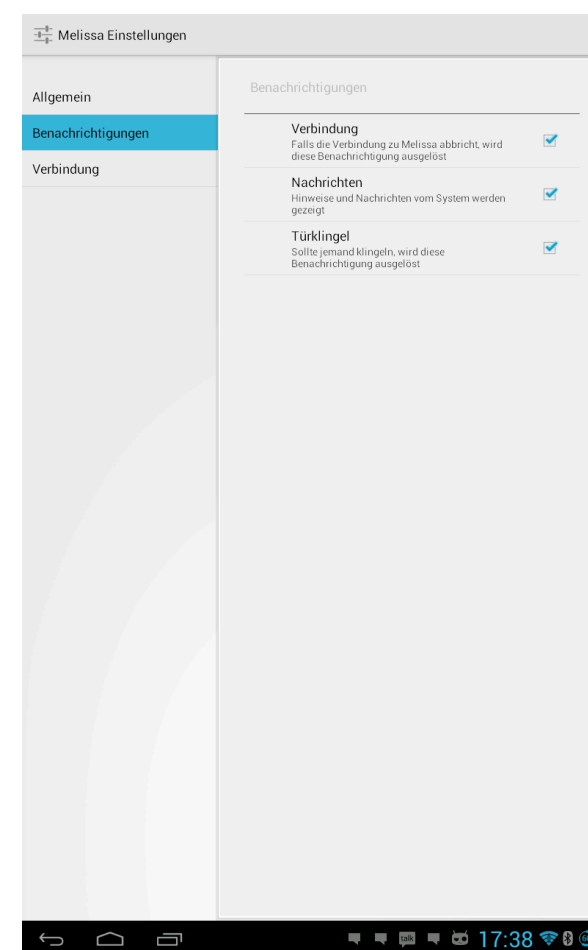
Screenshots



Widgetansicht



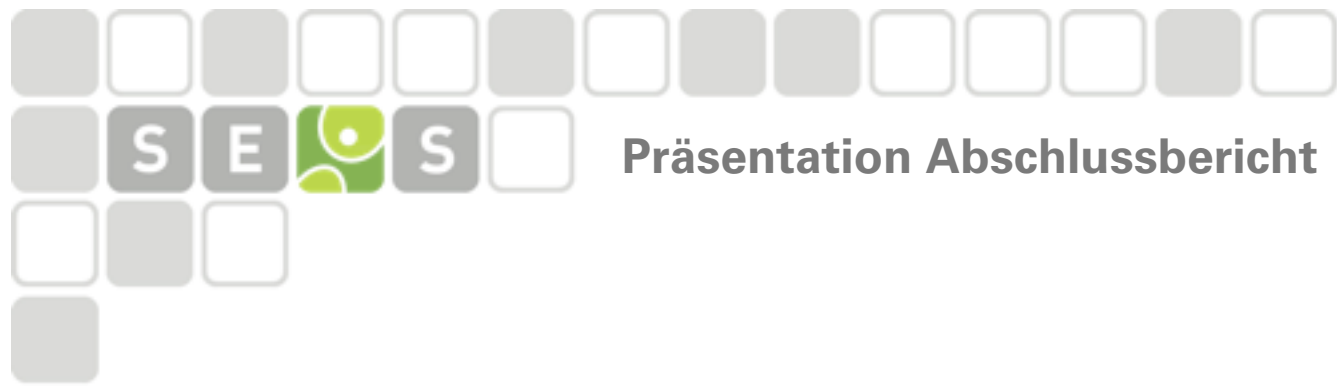
Sensorauswahl



Einstellungen

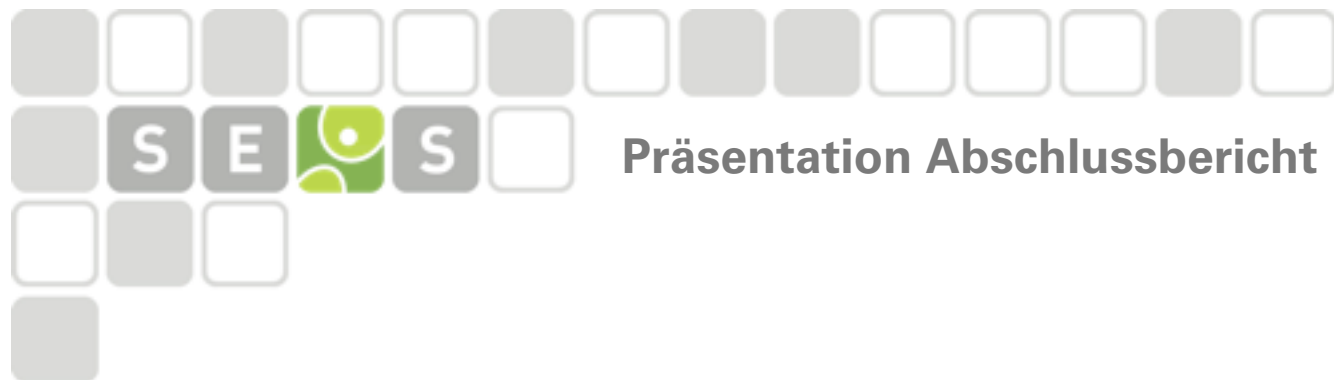
Probleme & Fazit

- Probleme
 - Kommunikation
 - Technisch
 - Zwischen den Gruppen
 - Praktikumsorganisation
- Fazit
 - Phase des Prototypings zu kurz
 - Gruppenabhängigkeiten störend
 - Androiderfahrungen vorausgesetzt



Zusammenfassung

- 13 Visualisierungen
- Parametrisierbarkeit
- Unterstützung vieler Aktoren und Sensoren
- Dynamische Anzeigerauswahl



Danke.