

User Stories als Basis für den Entwurf einer Architektur für ein flexibles System zur Visualisierung von Informationen im Smart Home

Autor: Philipp Jenke
Stand: 17. September 2015

1. Abgrenzung

Die Architektur beinhaltet alle für die Visualisierung notwendigen Bausteine. Sie setzt sich aus einer Persistenz-Schicht, einer Logik-Schicht und einer Darstellung (GUI) zusammen. In der Logik-Schicht wird ein Stapel von Ebenen (Layer) verwaltet. Jeder Layer dient einer spezifischen Informationsvisualisierung (z.B. Grundriss, Temperatursensoren, Zustände der Fenster (auf/zu), ...). Nicht enthalten in dieser Architektur sind der physikalische Aufbau des Smart Homes (z.B. Sensoren und Aktoren) und der Zugriff/das Auslesen/das Steuern der Aufbauten. Dennoch gibt es über die Persistenz-Schicht natürlich eine enge Verzahnung mit dem Aufbau, da beispielsweise die Sensoren ihre Messwerte an die Persistenz übergeben. Die Visualisierung in der Logik-Schicht wiederum greift auf die Daten zu, um sie darzustellen.

2. Schichten der Architektur

2.1. Persistenz

Ein Smart Home setzt sich auch verschiedenen Informationen zusammen. Immer enthalten sind ein Grundriss und eine Menge von Meta-Informationen. Beispiele für Meta-Informationen sind Name, Standort, Typ, ... Zu einem Smart Home existieren weitere Informationen, die beispielsweise von Sensoren im Smart Home aufgenommen wurden. Alle Informationen zu einem Smart Home können von der Persistenz-Schicht abgefragt werden. Es können mehrere Smart Homes in der Persistenz-Schicht abgelegt sein.

Die Persistenz-Schicht bietet mehrere Schnittstellen nach außen. In jedem Fall notwendig ist eine API über die alle Funktionen angesprochen werden können. Außerdem kann mit der Logik über eine Webschnittstelle (REST) kommuniziert werden. Beide Schnittstellen müssen detailliert dokumentiert sein.

2.2. Logik

Aus der Logik-Schicht heraus kann ein Smart Home geladen werden. Es können beliebig viele Ebenen (Layer) eingefügt werden. Die Layer sind selbst dafür verantwortlich, eine Visualisierung zu erzeugen. Die Logik-Schicht sorgt lediglich für die Ansteuerung dieser Aufrufe. Die Darstellung ist zeitabhängig. Die Logik hat immer einen aktuellen Zeitpunkt, den sie an die aktiven Layer übermittelt. Diese wiederum passen ihre Darstellung diesem Zeitpunkt an. Die Logik bietet mehrere Schnittstellen nach außen. In jedem Fall notwendig ist eine API über die alle Funktionen angesprochen werden können. Außerdem kann mit der Logik über eine Webschnittstelle (REST) kommuniziert werden. Beide Schnittstellen müssen detailliert dokumentiert sein.

2.2.1. Layer

Jeder Layer in der Lage, einen Szenengraphenknoten zu generieren (`cgresearch.graphics.scenegraph.CgNode`). Für die Darstellung von Szenengraphenknoten wird der gleiche Ansatz gewählt, der bereits im bestehenden Framework Anwendung findet: Ein Rendersystem (z.B. JOGL) bekommt den gesamten Szenengraphen. Es verfügt über Fabriken (Factories), die für jeden Szenengraphenknoten prüfen, ob sie dessen Inhalt (*content*) darstellen können. Falls ja, erstellen sie die notwendige Render-Repräsentation (siehe auch Kapitel 4.3 in der Dokumentation des Frameworks). (Beispiele für die zeitabhängige Darstellung: Bei der Darstellung eines Sensorwerte wird immer der Wert verwendet, der zum entsprechenden Zeitpunkt vorlag, die Darstellung des Grundrisses ist wahrscheinlich zeitunabhängig. Jeder Layer holt sich seine Daten primär aus der Persistenz-Schicht. Es ist aber auch denkbar, dass gemeinsam verwendete Funktionalität zum Zugriff auf Daten als Service in der Logik-Schicht umgesetzt und von den Layern verwendet wird.

2.3. Darstellung (GUI)

Die GUI zeigt den aktuellen Zustand der Visualisierung an. Dies sind insbesondere die aktuell geladenen Layer und deren Eigenschaften. Außerdem gibt es Zugriff auf die Meta-Informationen zum aktuell geladenen Smart Home. Über die GUI kann ein neues Smart Home geladen werden. Über die GUI kann auch der aktuelle Darstellungszeitpunkt eingestellt werden (z.B. über einen Schieberegler).