

Frank@Schlesinger.com

ElectroCodeoGram

Hintergrund & Architektureinblick

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- Aktueller Stand
- Ausblick

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- Aktueller Stand
- Ausblick

Aufgabe

- Thema: „Protokollierung von Programmieraktivitäten in der Eclipse-Umgebung“
- Für einen guten Zweck :)
- Name: ElectroCodeoGram (ECG)
- Thema alternativ: „Baue ein Messwerkzeug für Sebastian“

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- Aktueller Stand
- Ausblick

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- Aktueller Stand
- Ausblick

Kontext

- Erkenntnisse über Auftreten von Defekten gewinnen
- Messen – Wie wird programmiert?
- Analysieren – Wo sind Defekte?
- Hoffen - Abhängigkeiten entdecken und quantifizieren können

Mikroprozess ...

- ... des Programmierens
- Feingranulare Abbildung der Tätigkeiten beim Programmieren auf *relevante* Ereignisse
- Welche Ereignisse sind relevant?

Episoden

- Abfolge von Mikroprozessereignissen (MPE), die im Verdacht stehen besonders häufig zu Defekten zu führen
- Trial-And-Error, Cut-And-Paste, Work-And-Interrupt-And-Work,

ElectroCodeoGram

- Sensoren zeichnen MPEs in verschiedenen Umgebungen auf
- Anfallende Daten werden adhoc analysiert
- Analyse durch laufzeitladbare Module
- MPEs laufen einen Modulgraphen entlang

ElectroCodeoGram

- Verschiedene Ausgabemodule
- Verschiedene Einlesemodule
- Modul API
- Leichte Installation und Bedienung

Architektur

Schwerpunkte

HackyStat & ECG

Modul Management

MPE Management

MPE Transport

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- Aktueller Stand
- Ausblick

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- Aktueller Stand
- Ausblick

HackyStat

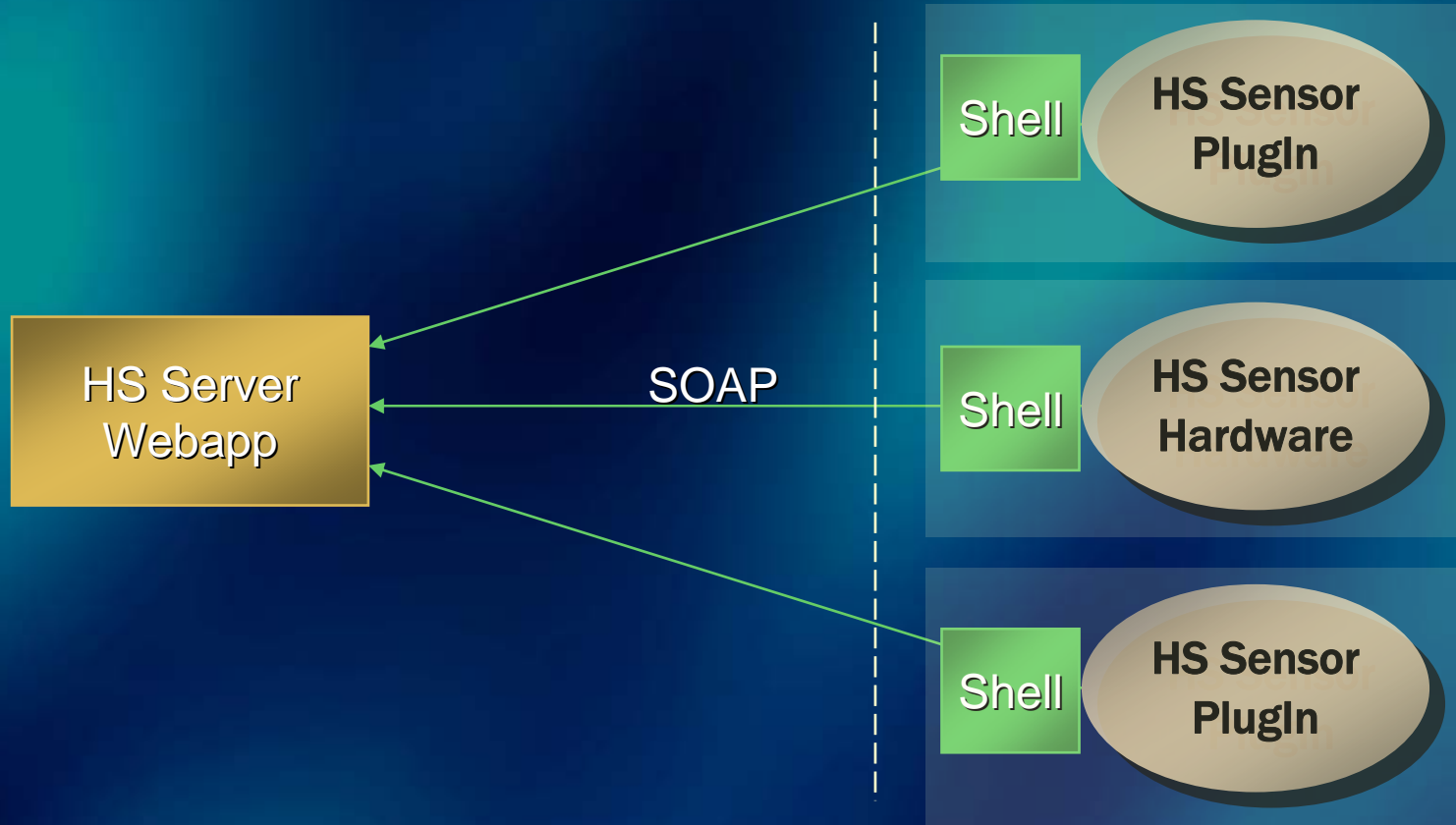
- Ist eine verteilte Anwendung zur Aufzeichnung und zentralen Speicherung von Ereignissen bei der Arbeit mit Anwendungsprogrammen
- HackyStat (HS) entwickelt von Prof. Johnson, University of Hawaii
- HS wurde über Jahre ständig weiterentwickelt (mehrmals Abriss-Neubau)

HackyStat

- Es existieren Sensoren (Eclipse, Emacs, Office, VS .NET, ...), die Daten aufzeichnen.
- Ein Server sammelt Daten aller aktiven Sensoren.
- HS ist dabei bzgl. der Sensoren und der aufzuzeichnenden Daten erweiterbar.
- HS muss irgendwie benutzt werden!

HackyStat Architektur

HackyStat Architektur



HackyStat & Anforderungen

- Aufzeichnung des Mikroprozesses
 - HS „SensorDataTypes“ sind Makroprozess
 - SDT zwar erweiterbar, aber sehr aufwändig
- Lokalität der Messdaten
 - Daten in HS nur auf generierten Webseiten des Servers, oder via http-Anfragen
 - Lokal nur Backup als serialisierte Objekte

HackyStat & Anforderungen

- Adhoc-Auswertung der Daten
 - HS hat keine Schnittstelle zur Datenauswertung
- Echtzeit & Effizienz
 - Daten werden nur selten und gebündelt transportiert
 - Transport via SOAP

HackyStat & Anforderungen

- Leichte Installation
 - HS Installation ist schwierig
- HS muss ergänzt werden!

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- Aktueller Stand
- Ausblick

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- **Architektur**
- Aktueller Stand
- Ausblick

HackyStat Ergänzungen

**HS Server
Ergänzung**

HS Proxyserver

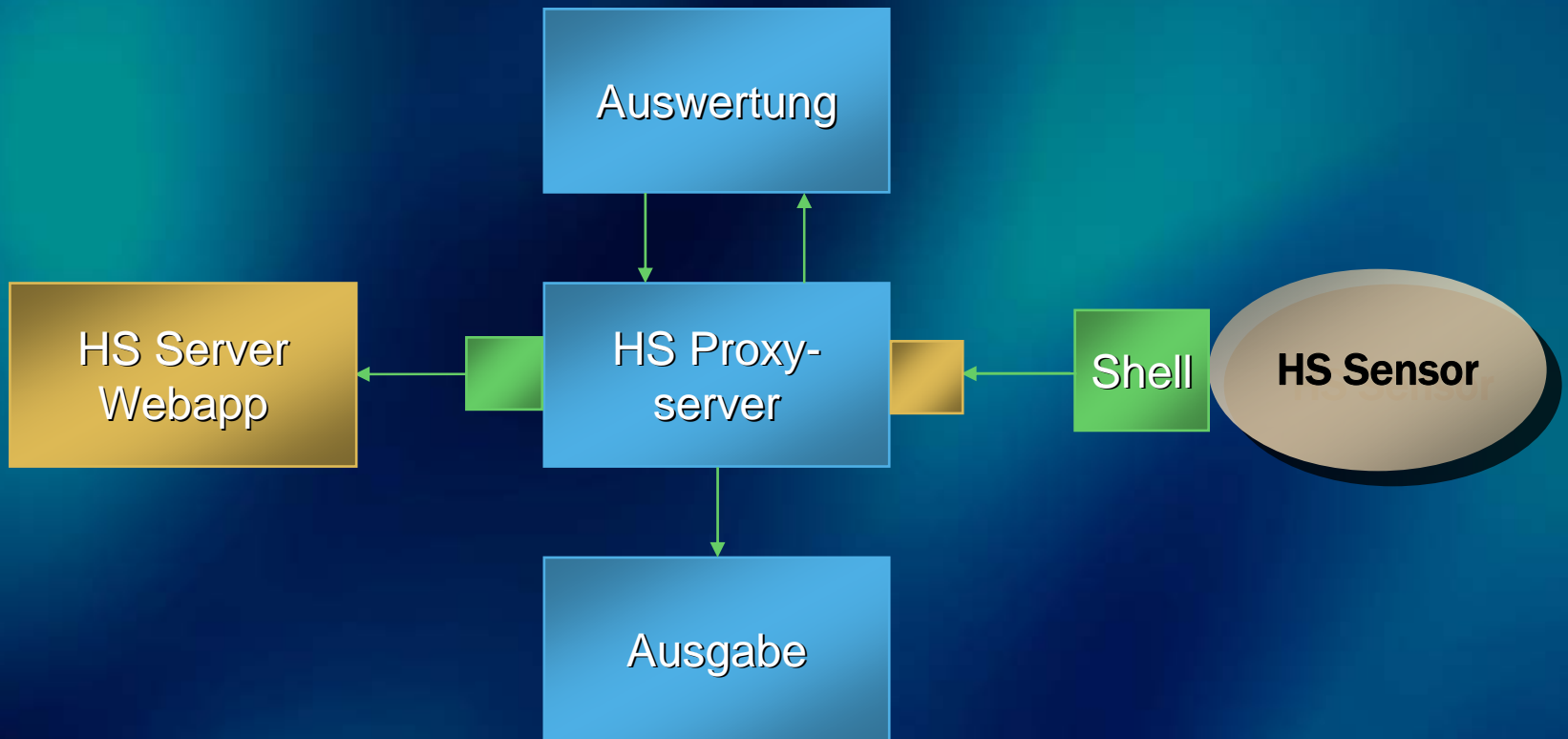
**HS SensorShell
umbauen**

Ansätze

HackyStat Server Erweitern

- Alle Daten fließen hier zusammen.
- Auswertung der Daten wäre machbar
- Probleme
 - Daten nicht lokal
 - Schwierige Installation
 - Keine Echtzeit
 - Ineffizienter Datentransport

HackyStat Proxyserver

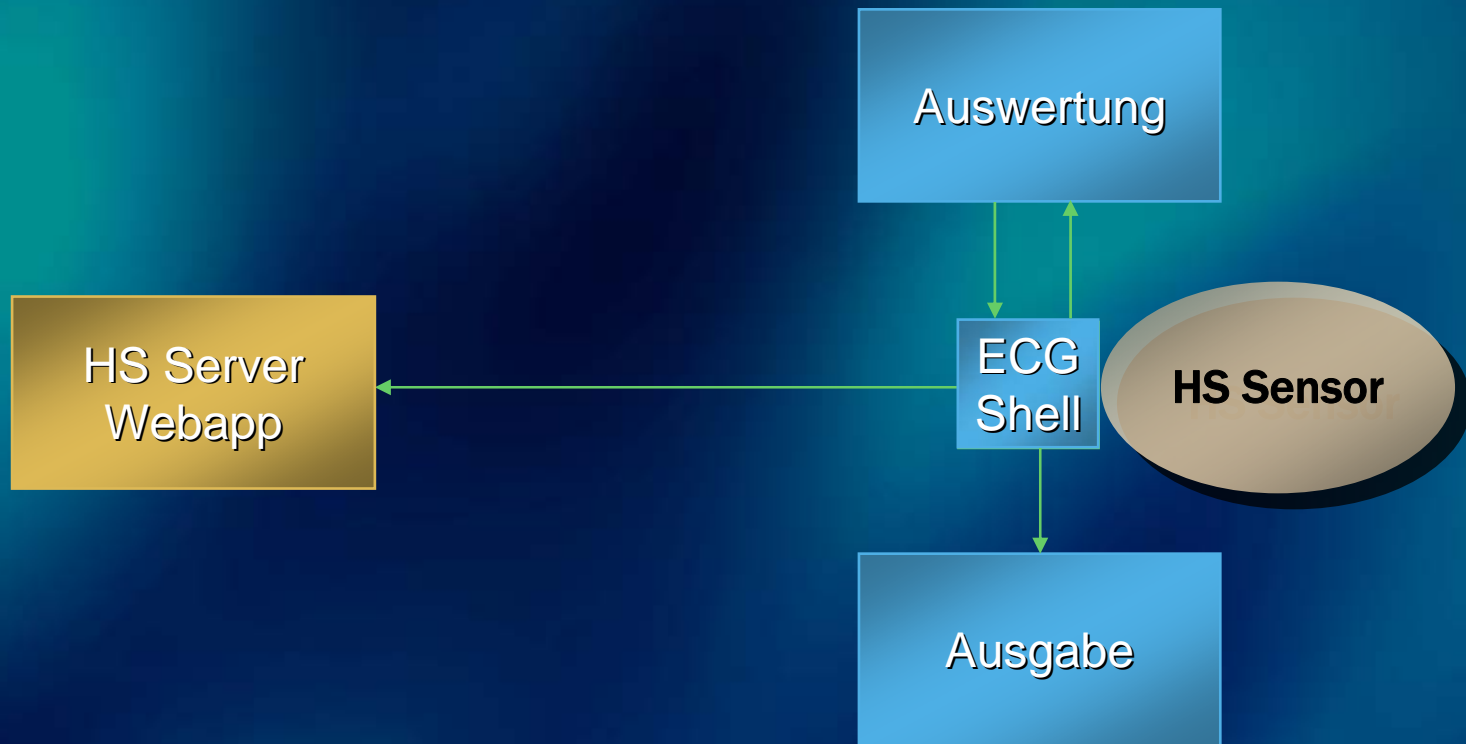


HackyStat Proxyserver

- Saubere Lösung
- Leichte Installation
- Probleme
 - Keine Echtzeit
 - Ineffizienter Datentransport
 - SDTs schwierig erweiterbar

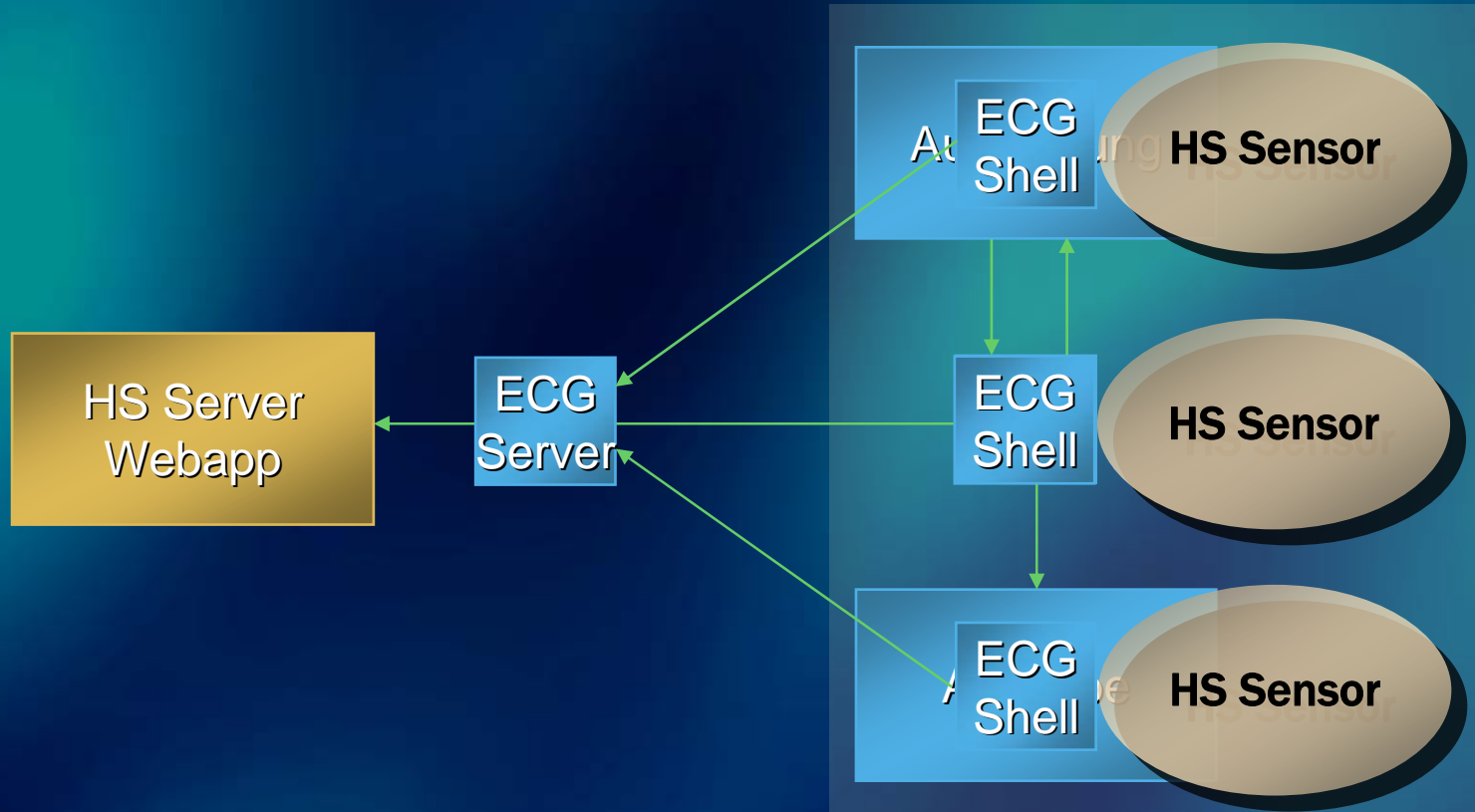
HS SensorShell umbauen I

Erweiterung der SensorShell



HS SensorShell umbauen II

Erweiterung der SensorShell



HS SensorShell umbauen

- Erfüllt die Anforderungen
- Ist bzgl. Sensoren, „SensorDataTypes“ und HS Server kompatibel
- Problem
 - Aufwändig
- Diese Architektur wurde implementiert.

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- **Architektur**
- Aktueller Stand
- Ausblick

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- **Aktueller Stand**
- Ausblick

Aktueller Stand

- Implementierung der Architektur beinahe abgeschlossen
- Momentan Qualitätssicherungsphase
- Funktionale Modultests
- Stresstests

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- **Aktueller Stand**
- Ausblick

Inhalt

- Aufgabe
- Hintergrund & Idee
- HackyStat
- Architektur
- Aktueller Stand
- **Ausblick**

Ausblick

- Implementierung der funktionalen Inhalte
 - Generische Episodenerkennung
 - Codechange Differ
 - VS .NET Sensor
 - Studienarbeiten
-
- Stay connected:
www.electrocodeogram.org



ElectroCodeoGram.org

AG SoftwareEngineering
Institut für Informatik
Freie Universität Berlin