



Erkenntnisse u. Architektur Pilotprojekt

Felix Korb (BTC), Dr. Matthias Rohr (BTC),
openKONSEQUENZ Workshop Frankfurt 28.10.2015 zur
Anfragevorbereitung neuer Module

Inhalt

- Vorgehen u. Erfahrungen
- Status u. Entwicklungsumgebung
- Beispielscreens u. -prozess
- Architekturüberblick
- Datenmodell
- Ausblick

Pilotprojekt - Vorgehen und Erfahrungen

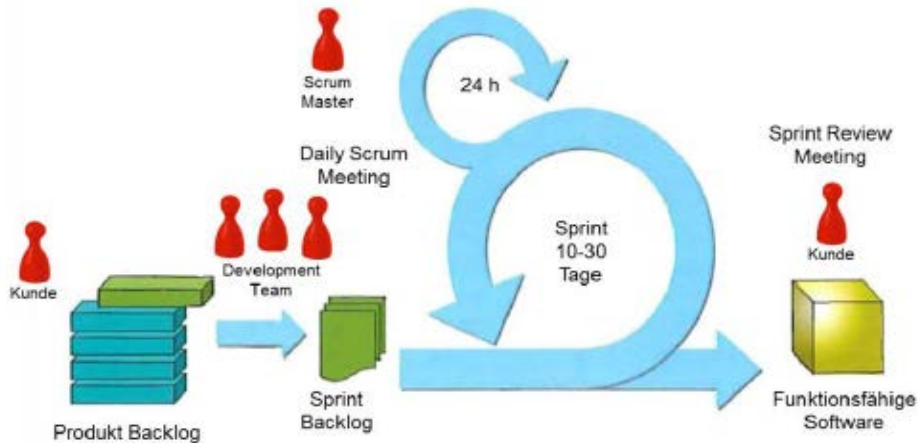


Bild: Basis Wikipedia: <https://de.wikipedia.org/wiki/Scrum> Lakeworks derivative work: Sebastian Wallroth, CC BY-SA 3.0

Erfahrungen

- vereinheitlichte Nomenklatur wichtig: im Energiesektor werden viele Begriffe mehrdeutig verwendet
- zeitintensive Lizenz-Überprüfung von Komponenten
- SCRUM muss speziell auf unternehmensübergreifendes Arbeiten ausgerichtet werden

SCRUM-basiertes Vorgehen

- täglicher Austausch zwischen Hersteller und Netzbetreiber
- kein klassisches Lasten- bzw. Pflichtenheft notwendig
- mitwachsende Anforderungen auf Basis von Epics/User Stories
- Epics/User Stories zu Projektanfang oder im Vorfeld initial vorabklären
- „agiles“ Erstellen von Teilergebnissen
- neue Lösungsansätze durch enge interdisziplinäre Zusammenarbeit

**Lessons Learned
Workshop**
10. November 2015

Pilotprojekt – Entwicklungsrahmen

openKONSEQUENZ ist...



Eclipse

...ein open-source
Eclipse Foundation
Projekt



EPL

...unter
Eclipse Public License
lizensiert



GIT

...im
Eclipse GIT
öffentlich verfügbar

Technologie:

- Back-End: Java, JPA, Spring, Postgres, ...
- Schnittstellen:
 - Front-End ↔ UI: REST, JSON
 - Im-/Export: CIM (über ESB)
- UI (webbasiert): AngularJS, Bootstrap, ...
- Build-Tool: Maven

Qualitätssicherungsmaßnahmen:

- automatische Style-Checks
- Code-Reviews
- Unit-Tests
- automatischer Build-Prozess
- klassische SW-Test im Rahmen von QS-Maßnahmen
- zeitnahe Tests durch Kunden

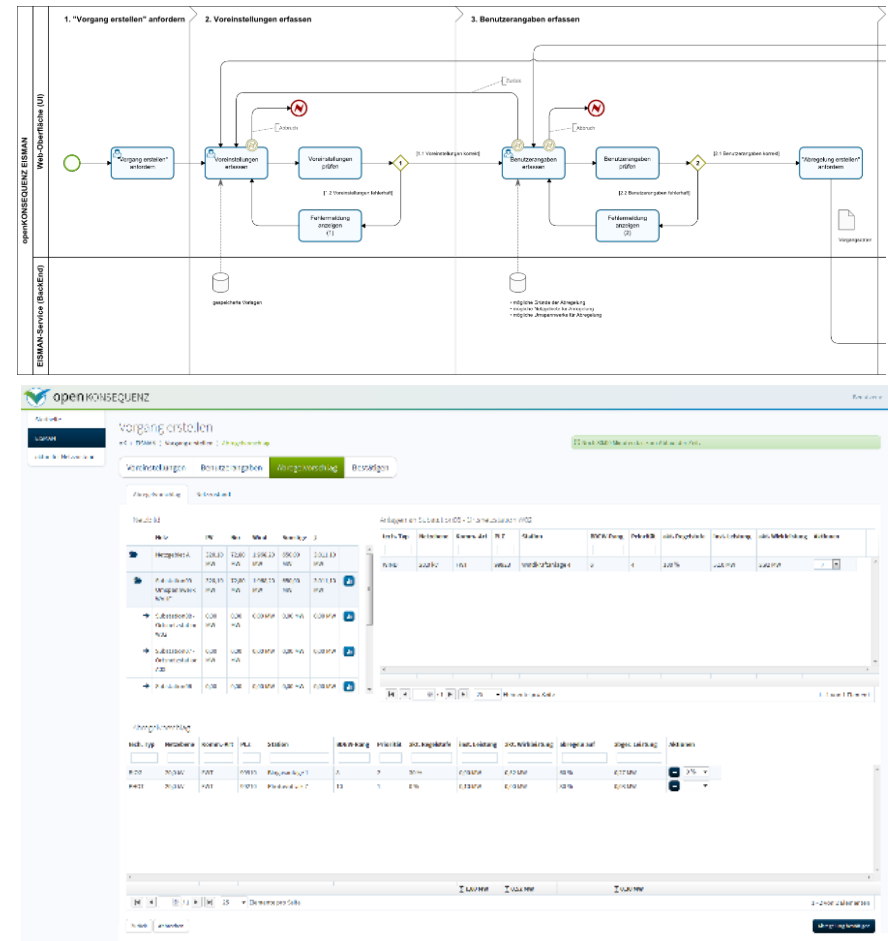
„Eclipse-Luna-Logo“ von Eclipse Foundation - <http://www.eclipse.org/artwork/>. Lizenziert unter Logo über Wikipedia

„Git-logo“ von Jason Long - <http://git-scm.com/downloads/logos>. Lizenziert unter CC BY 3.0 über Wikimedia Commons

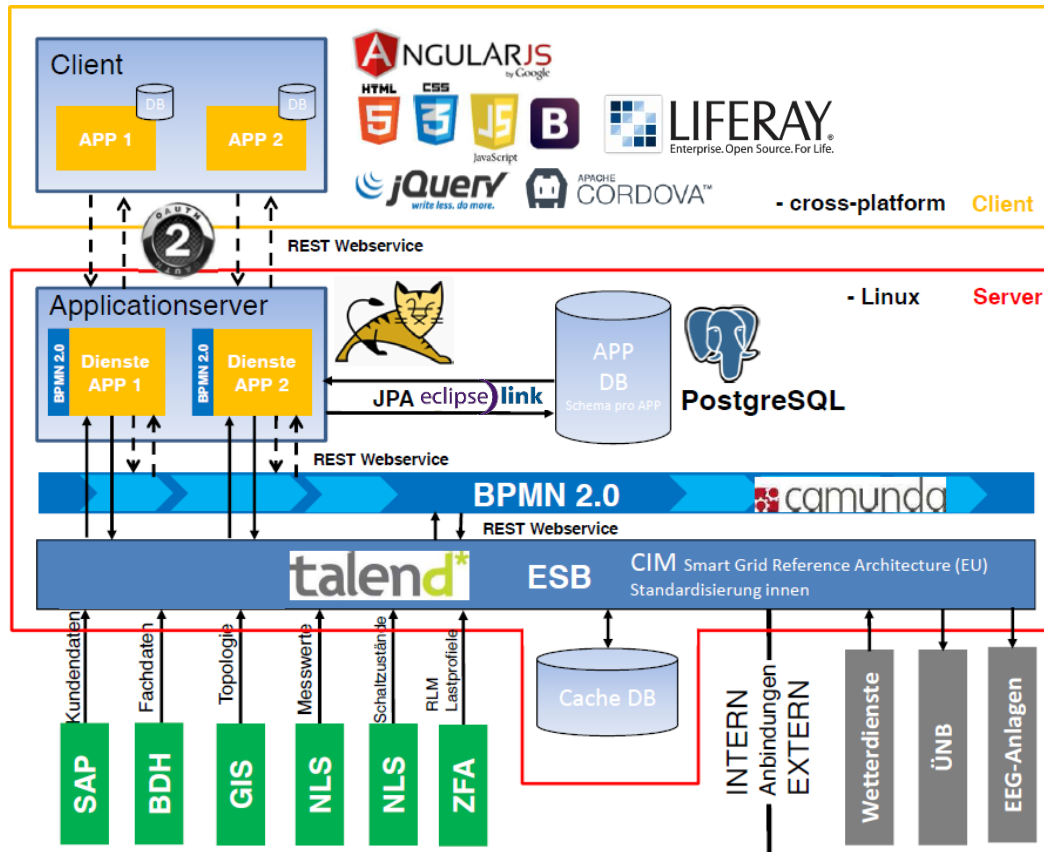
Pilotprojekt

aktueller Entwicklungsstand

- Anforderungen sind erfasst
- User-Stories in Prozesse überführt
- Prozesse abgestimmt
- Oberflächen abgestimmt
- regelmäßige Auslieferung auf Referenzplattform (Darmstadt)
- Entwicklung ist in der Phase der letzten Sprints



Technologieeinsatz im Pilot im Vergleich zur Ursprungsarchitektur



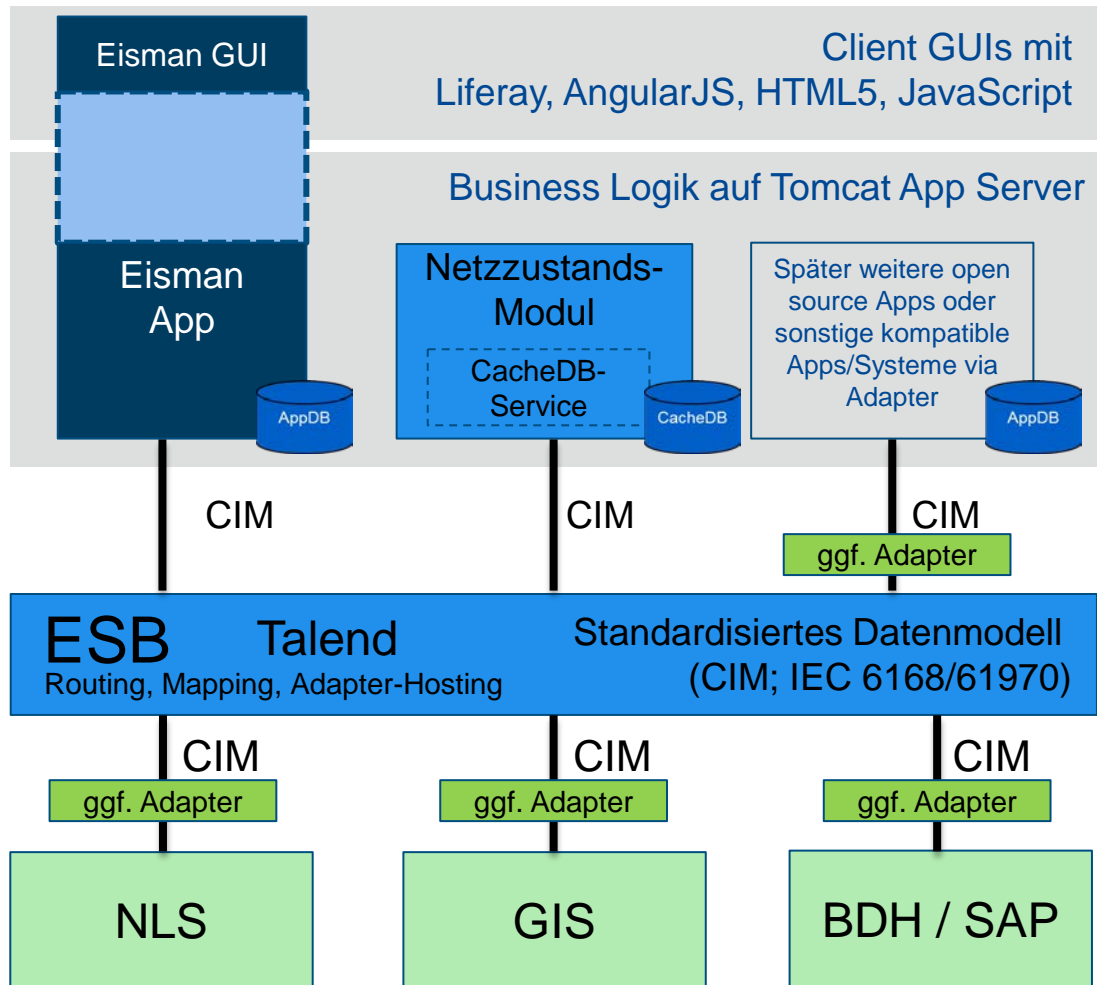
Einsatz im Pilotprojekt u. in Referenzplattform im Vergleich zur ursprünglichen Architektur:

Portal: Liferay
 ESB: Talend
 ORM: Eclipselink
 App-Serv.: Tomcat

(Andere DBMSs, ESBs, Java-App-Server möglich)

Bisher nicht genutzt, da für Pilot nicht relevant:
 Cordova,
 Camunda

Architekturkonzept openK Plattform mit Eisman-Modul im Überblick



Eisman-App + GUI:

- Fachlogik für Erstellung Abschaltvorschlag
- Pilotfokus auf Eisman-Systemsicherheitsmanagement (Typischer ÜNB-Abruf)
- Web-Oberflächen zur Bedienung
- Nutzt Netzzustandsmodul als integriertes Datenmodell (z.B. welche Einspeiser hängen an UW Trafo)

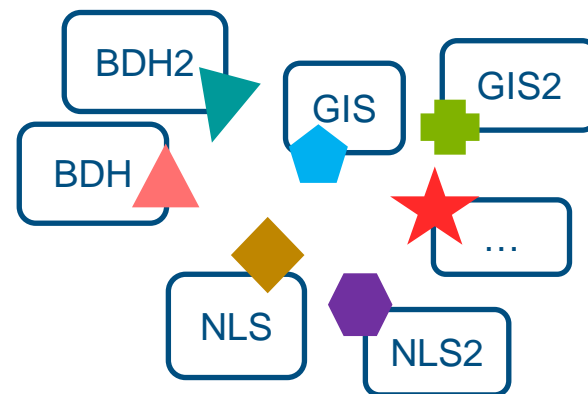
Netzzustands-Modul inkl. CacheDB-Service + CacheDB

- Verwendet CIM-basiertes integriertes Datenmodell für Topologie, Messdaten, Stammdaten
- Derzeit auf Eisman fokussiert; Bedarfsgetriebene inkrementelle Erweiterungen
- Teil der openK Plattform, da Nutzenpotential für viele Apps
- Kompensation der geringeren Verfügbarkeiten der Dritt-Systeme u. Erhöhung der Performance

Interoperabilität und Integrationsfähigkeit durch Standards

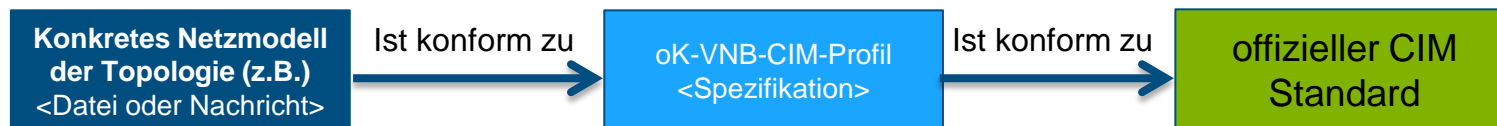
Status Quo:

- Stark heterogenes Datenmodell in der VNB-Systemlandschaft
- Proprietäre Schnittstellen
- ➔ Hohe Integrationshindernisse u. –kosten bremsen sowohl VNBs als auch Hersteller aus (Doppelprojektierung, Inkonsistenzen, Integrationsaufwände, ...)

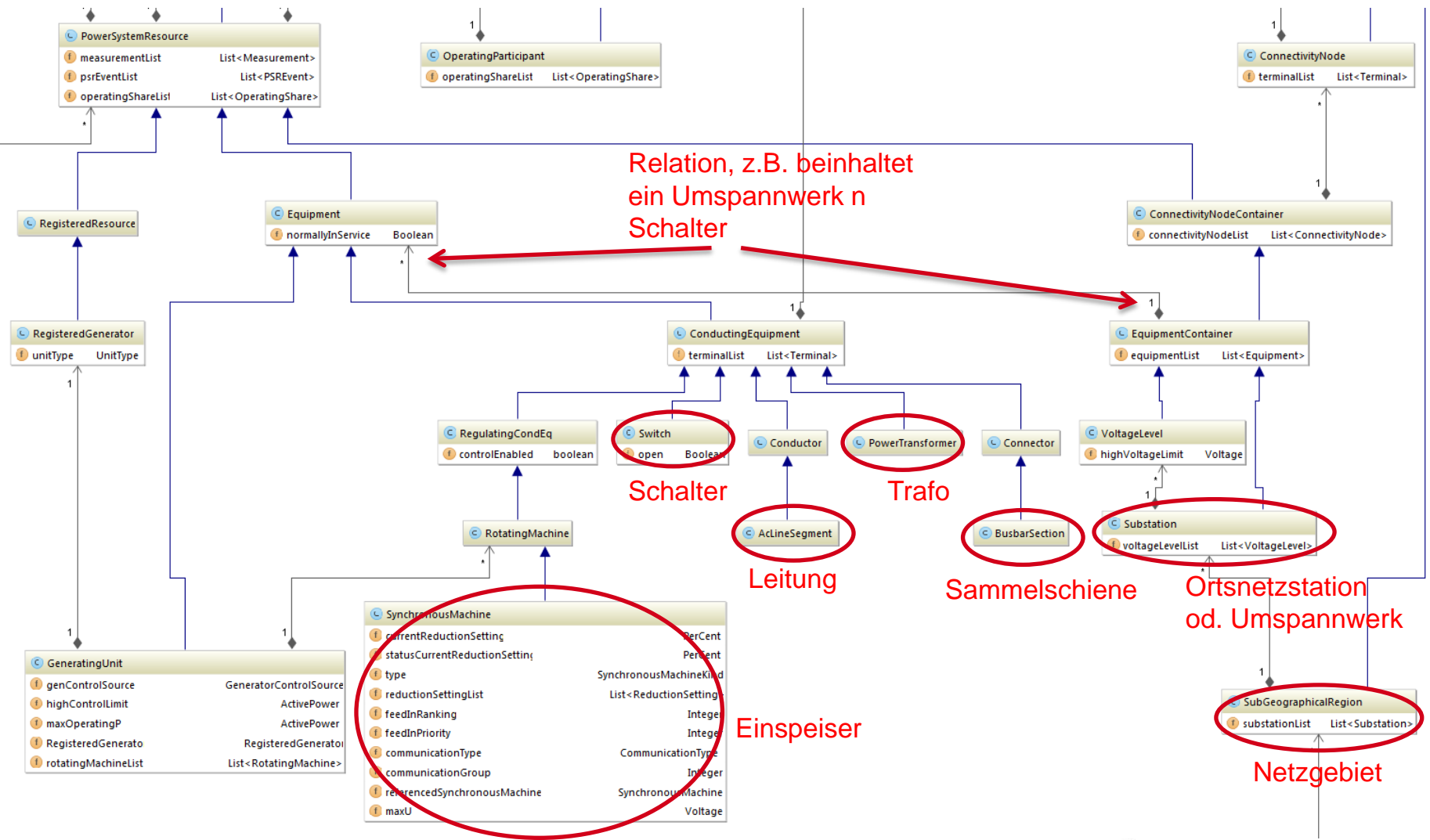


Ansatz / Vision:

- Offene Schnittstellen/Datenmodell auf Basis des CIM Standards (IEC 61968 / IEC 61970)
- Konkretisierung durch Abstimmung durch openKONSEQUENZ; ggf. analog zu ENTSO-E Profil



Beispiel-Modellausschnitt aus Pilotprojekt



Beispiel: CIM - Erweiterungen für openk platform

Neue Attribute in SynchronousMachine (Einspeiser)

- reductionSettingList (mögliche Absenkleistung einer Anlage)
- currentReductionSetting (aktuelle Absenkleistung)
- statusCurrentReductionSetting (Status)
- feedInRanking (Einspeiseranking nach BDEW)
- feedInPriority (Einspeisepriorität des Netzbetreibers)
- communicationType (FWT, EFR, ...)
- communicationGroup (z.B. Rundsteuergruppe mehrerer Anlagen, die nur gemeinsam geregelt werden können)
- referencedSynchronousMachine (Referenzanlage)

Aktueller Stand bzgl. CIM Datenmodell und Ausblick

Stufe 1 (aktuelles im Pilot):

- Integriertes Datenmodell (statische + dynamisch Topologie + Stammdaten + Messwerte) mit Fokus Einspeisemanagement
- Fokus auf CIM-basierte Semantik im Datenmodell (Beschrieben als UML-Modell)

Zusätzlich in einer Stufe 2:

- Erweiterung der *semantischen* CIM Interoperabilität auf volle *technische* CIM Standard/Praxis-Konformität (insb. XML u. RDF)

Zusätzlich in späteren Stufen denkbar:

- Ggf. vergleichbare Spezifikation und Praxis zu CIM-entso-e-Profil
- Ggf. Rückkopplung von VNB-spezifischen, aber allgemein-gültigen openK-CIM-Erweiterungen in internationalen CIM-Standard

Weiterer Architekturworkshop geplant

- Separater Termin mit Ziel den Stand der Architektur und openK Platform im Detail zu vermitteln
- Insb. auch als Fragetermin gedacht
- Videokonferenz + Telefontelko (Microsoft Lync + Telefontelko)
- Termin voraussichtlich Mitte Januar
=> Rechtzeitig vor den Ausschreibungen
- Einladung wird vorab über den üblichen Eclipse-Verteiler verschickt, der auch für diesen Termin genutzt wurde (+Vortermine zum Testen der VideoConf-Technik)



**Vielen Dank für ihre
Aufmerksamkeit!**