SOA @ T-Mobile – SOA und Subversion? Wir zeigen den Zusammenhang.

Subconf 2008, München, Carsten Sensler & Andre Karalus

Agenda

- SOA @ T-Mobile – The SOA Backplane program
- SOA Backplane – Zoom into ESB
- Why integration with Subversion?

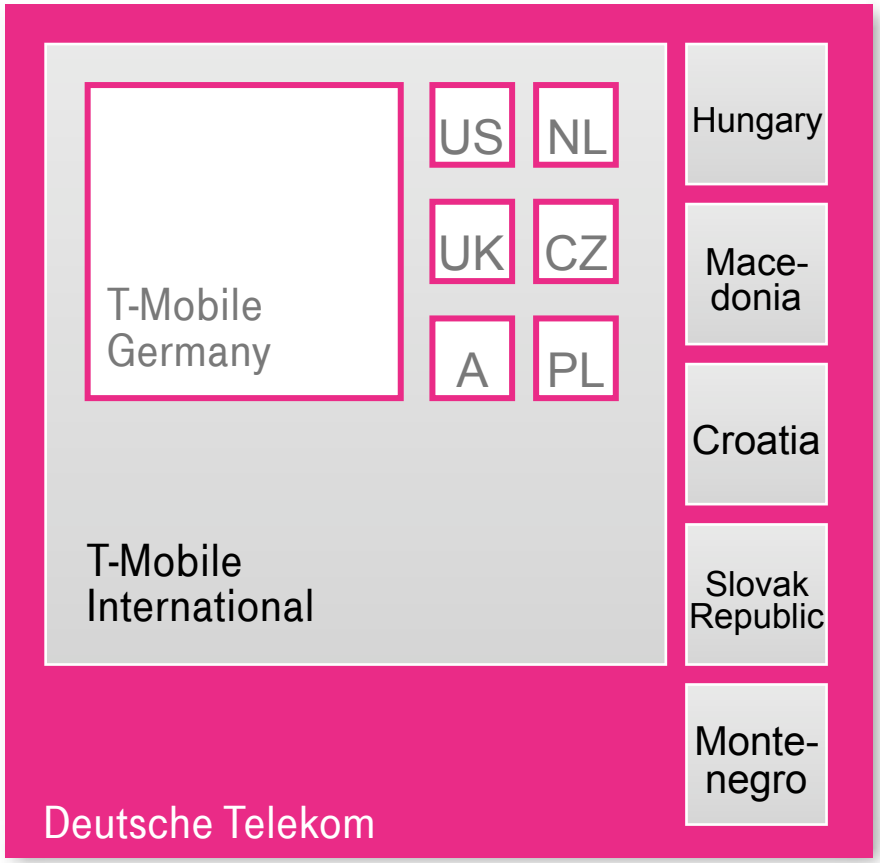
Who we are?

- Dipl.-Ing. Carsten Sensler
  - Employee of T-Mobile Deutschland GmbH since April 2007 (but since December 2005 working in the SOA Backplane program)
  - Department of Enterprise Integration
  - System & Solution Designer
  - Functional teamleader of the international Service Provisioning Team

- Dipl.-Inf. Andre Karalus
  - Freelancer,
  - since March 2006 consultant for T-Mobile
  - Designer and developer of the runtime core component of the ESB from the SOA Backplane and of the Core from the Service Repository

SOA @ T-Mobile –
The SOA Backplane program

Corporate Structure.
Subsidiaries and affiliates.



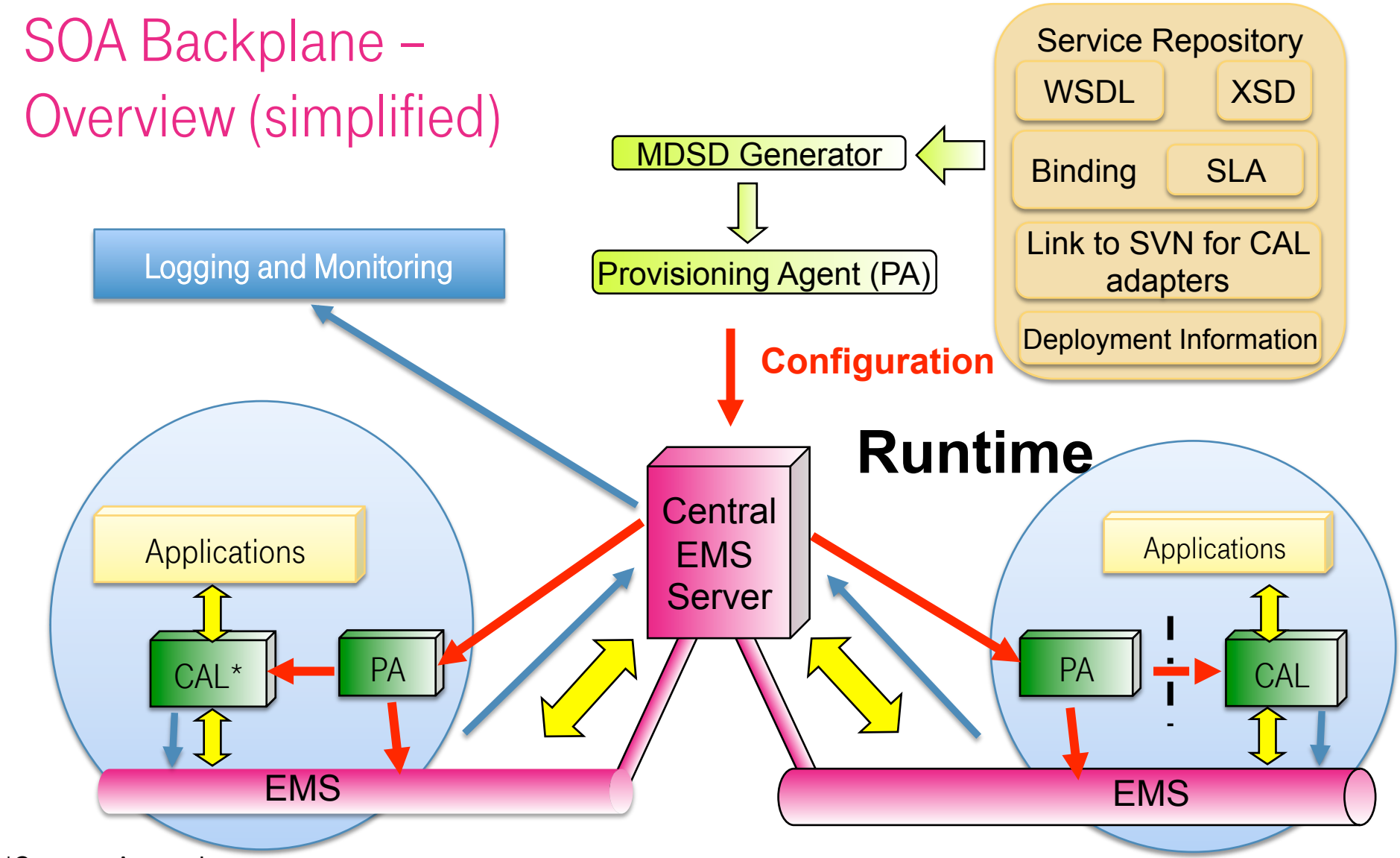
- Direct or indirect investments by Telekom Group in companies dealing with mobile communications in twelve countries
- The T-Mobile brand is represented in Germany, Austria, Hungary, Great Britain, the Czech Republic, the Netherlands, the Slovak Republic, Croatia and the USA
- Almost 125 million customers in the majority holdings

SOA Backplane Participant

Intention of the SOA Backplane program

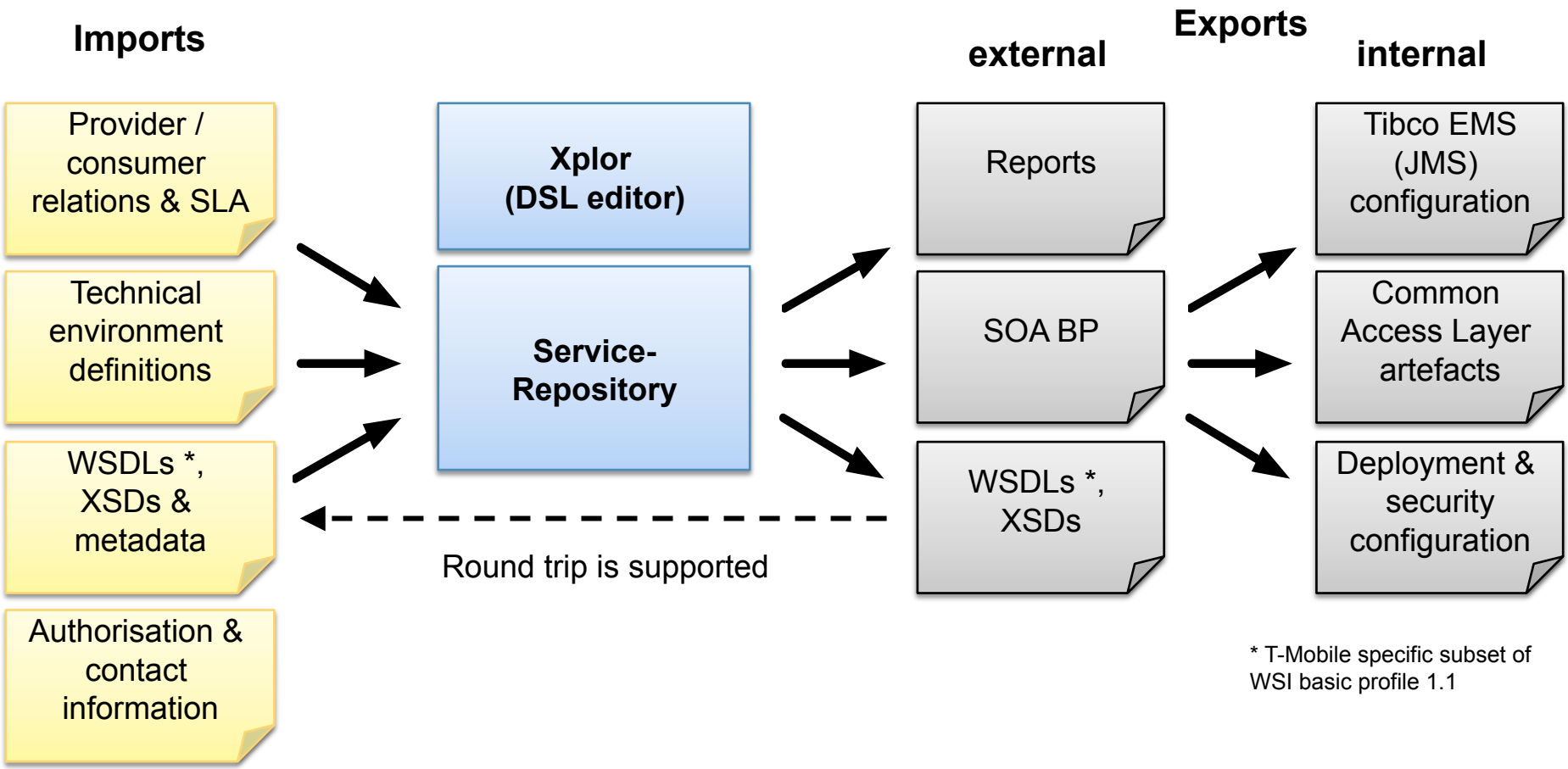
- SOA Backplane will deliver a number of software systems and standards, namely
  - a service bus which is the SOA communication infrastructure
    - Service repository
    - Access layer framework
    - Basic messaging infrastructure (JMS)
  - additional value adding components and functionality including
    - logging, monitoring
    - service contract management
    - business activity monitoring
    - transport components for B2B communication
  - the Backplane Guide and SOA Governance as a set of guidelines and rules as to how SOA will be implemented within T-Mobile.

SOA Backplane – Overview (simplified)



*Common Access Layer

SOA Backplane – Service Repository Interfaces



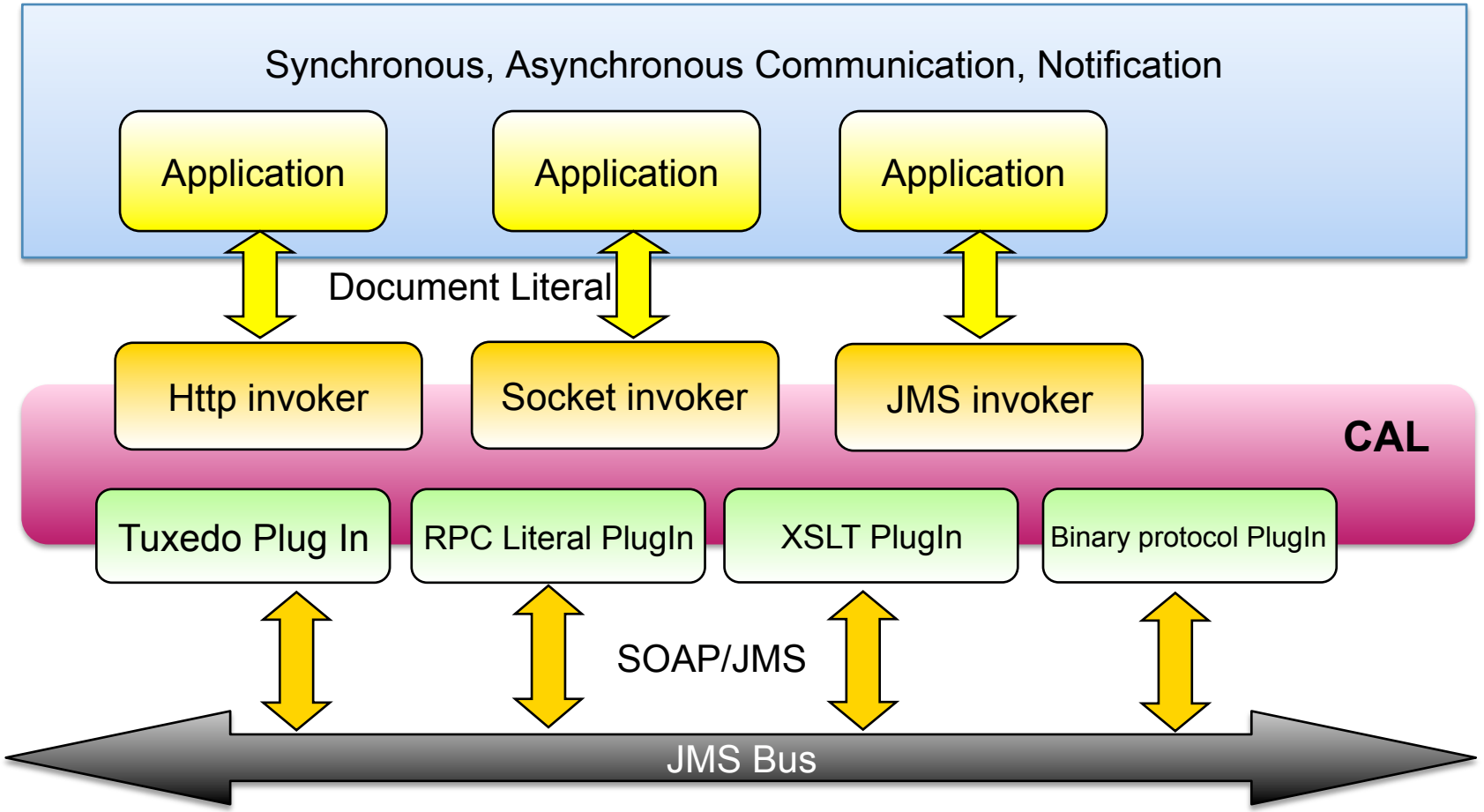
* T-Mobile specific subset of WSI basic profile 1.1

The Service Repository in a nutshell

- Provide all information needed by service participants for consistent service implementation and utilisation (architecture)
- Store definition of different SOA backplane environments and binding of service participants to these environments (binding)
- Support fully automated configuration of SOA backplane environments (dev, test, prod, ...) (service provisioning)
- Support SOA governance and impact analysis (change / incident) (management)

SOA Backplane – Zoom into ESB

SOA Backplane – Common Access Layer (CAL) - ESB



Protocols and Adapters

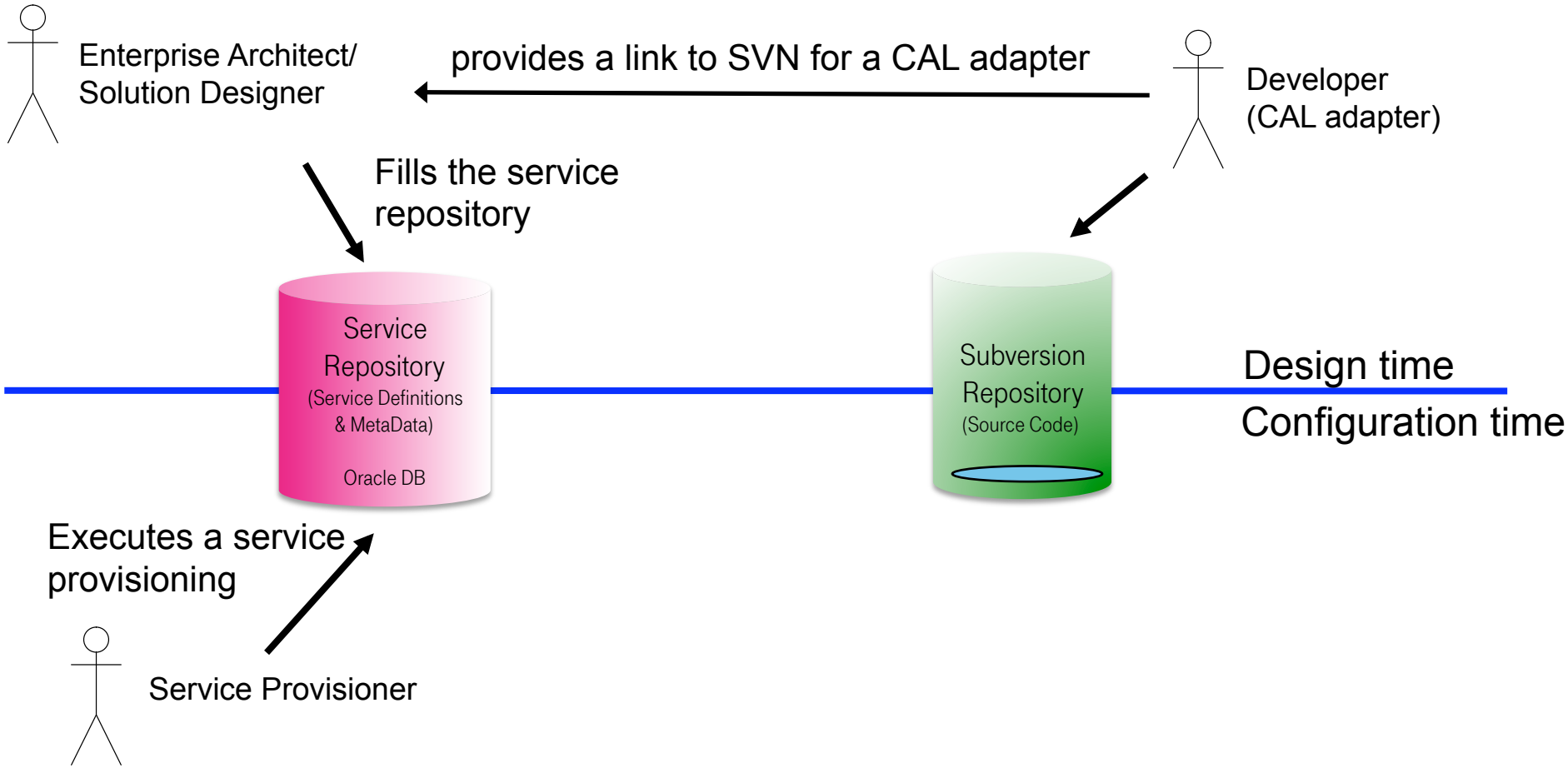
- The origin protocol on SOA BP is SOAP/http (document literal wrapped) and internally SOAP/ JMS
- With the Adapter/Plug-Ins one can translate the content of a message, e.g. rpc literal style to document literal wrapped, or even a proprietary binary protocol to SOAP
- With the protocol layer it is possible to use other protocols than http, e.g. direct socket or JMS
- The service specific adapter/plug-in code (java, XSLT or Groovy) is considered to be part of the configuration and not of the SOA BP core itself
- The adapter/plug-in code has to be subjected to versioning since it belongs to its according version of service definition

SOA Backplane – Service Provisioning

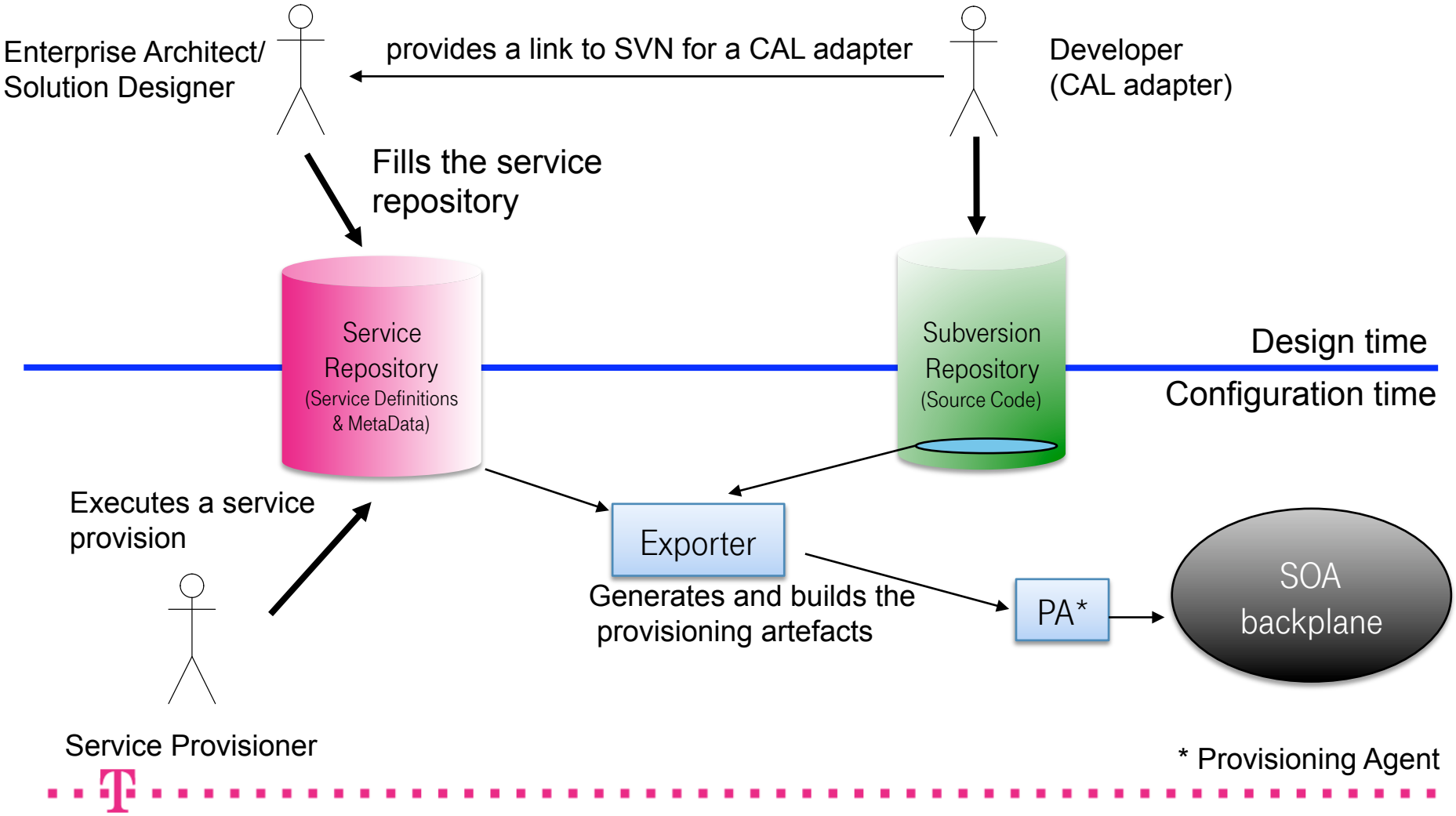
- Since we use static routing in order to have better control and insight into the communication on SOA BP, the information who can talk to whom needs to be propagated to the runtime components
- The CAL is provided by the deployment agent with the necessary configuration about the possible routings, the used message exchange patterns and quality of services parameters
- The EMS Server is configured by the deployment agent, too. It enables the messaging transport via queues
- Along with this configuration it is possible to propagate necessary adapter configurations, transformation scripts (e.g. XSLT) or even code that has to adhere to a strict interface

Why integration with Subversion?

Roles within the SOA backplane – “Design time” and “Configuration time”



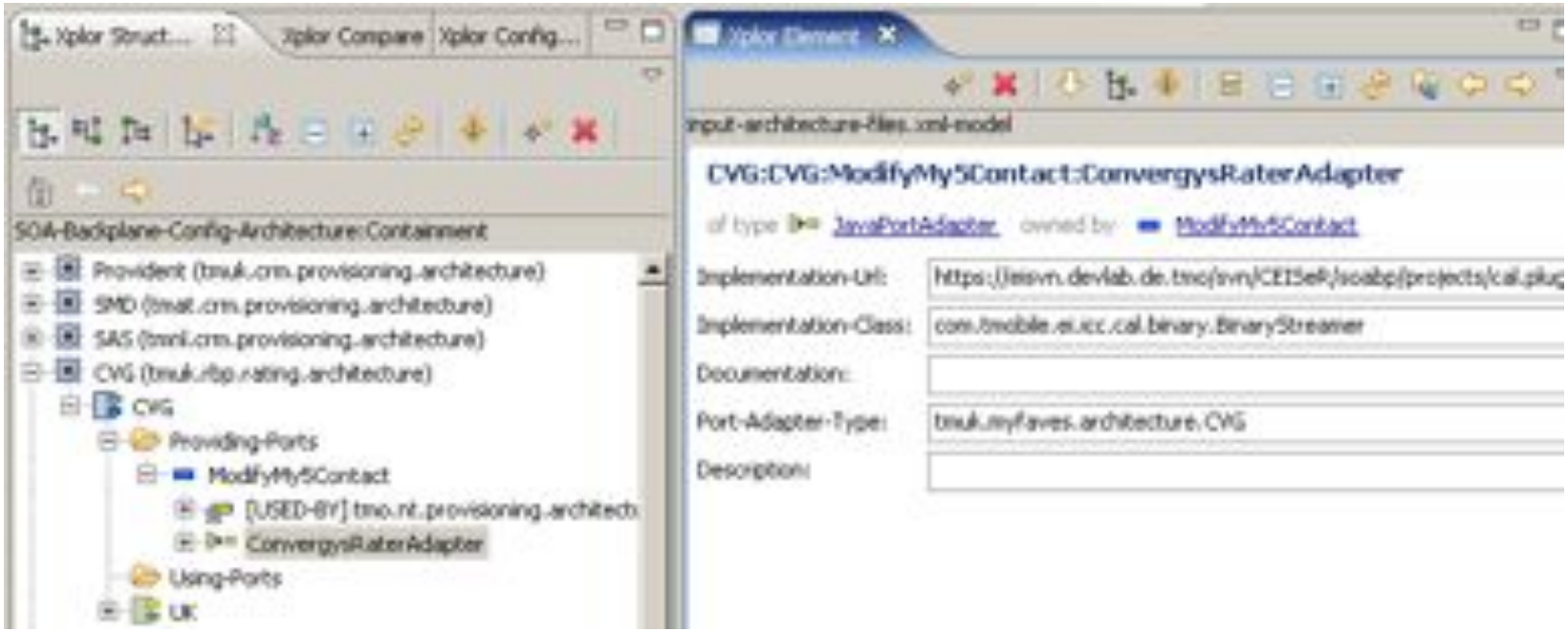
Service Provisioning and Subversion (1/3)



Service Provisioning and Subversion (2/3)

- A Java process based on ant (the exporter) combines the information out of both repositories into one configuration document that itself is a SOAP document
- Ant is used to build EARs that provide for a standardized (JEE-MDBs) way to access queues on an JMS-Severer
- The exporter utilizes svnant to access subversion
- The version and URI of the adapter/plug-in code is provided by the service repository

Service Provisioning and Subversion (3/3)



Additional use-cases of Subversion at T-Mobile

- Versioning of source code
- Versioning of the development infrastructure for a standardized development tool set (Eclipse incl. Plugins, JDK, JBoss, EMS, 3rd party tools such as test tools, etc.)
  - See track tomorrow morning ☺
- Versioning of documents
  - Requirements
  - Functional specifications
  - Operational manuals
  - Release plans

Why Subversion?

- No license charge, it's more difficult to agree on software that generates licensing costs between different stakeholders
- Excellent integration in our development process and environment
- Subversion is international reachable within the T-Mobile. It has the firewall clearances for all other NatCos and a connectivity to a central LDAP for authorization.
- With subversion, you can easily merge and branch
- Subversion can version directories
- Subversion has got a very high performance (@T-Mobile)

Links and Literatur

- Subconf 2007, C. Sensler & Andre Karalus: Integration eines SOA Repository mit Subversion zur Anbindung an den ESB der T-Mobile

- Javamagazin 10/08 – 12/08, C. Sensler & A. Karalus: SOA @ T-Mobile: Vollautomatische Service Provisionierung auf dem ESB

Discussion

Thank you.