

Standardschnittstellen als nichtfunktionale Variationspunkte: Erfahrungen aus der EPM-Produktlinie

Martin Frenzel, Jörg Friebe, Simon Giesecke, Till Luhmann

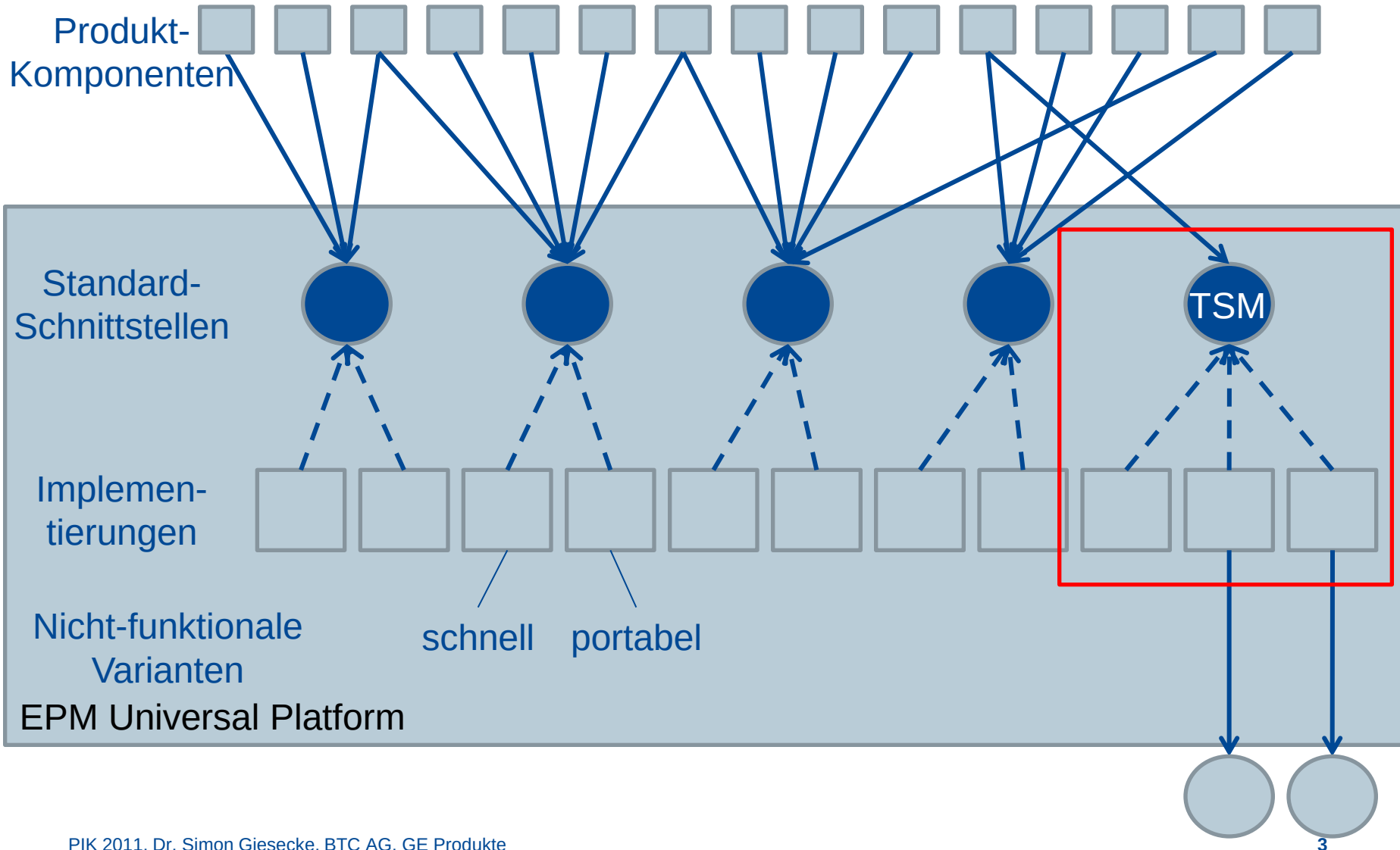
BTC AG, Standorte Oldenburg & Berlin

PIK2011, Karlsruhe, 22. Februar 2011



- **Motivation**
- **EPM-Produktlinie**
 - Überblick
 - EPM-Referenzarchitektur
 - EPM-Architekturstil
- **Architekturstandpunkte im EPM-Architekturstil**
- **Standardschnittstellen im Architekturstandpunkt Logische Softwarestruktur**
 - Schnittstellenarten
 - Nutzung von Standardschnittstellen zur Umsetzung nicht-funktionaler Variationspunkte
- **Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen**
- **Zusammenfassung & Ausblick**

Plattformbausteine der EPM Universal Platform



- **Motivation**
- **EPM-Produktlinie**
 - Überblick
 - EPM-Referenzarchitektur
 - EPM-Architekturstil
- **Architekturstandpunkte im EPM-Architekturstil**
- **Standardschnittstellen im Architekturstandpunkt Logische Softwarestruktur**
 - Schnittstellenarten
 - Nutzung von Standardschnittstellen zur Umsetzung nicht-funktionaler Variationspunkte
- **Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen**
- **Zusammenfassung & Ausblick**

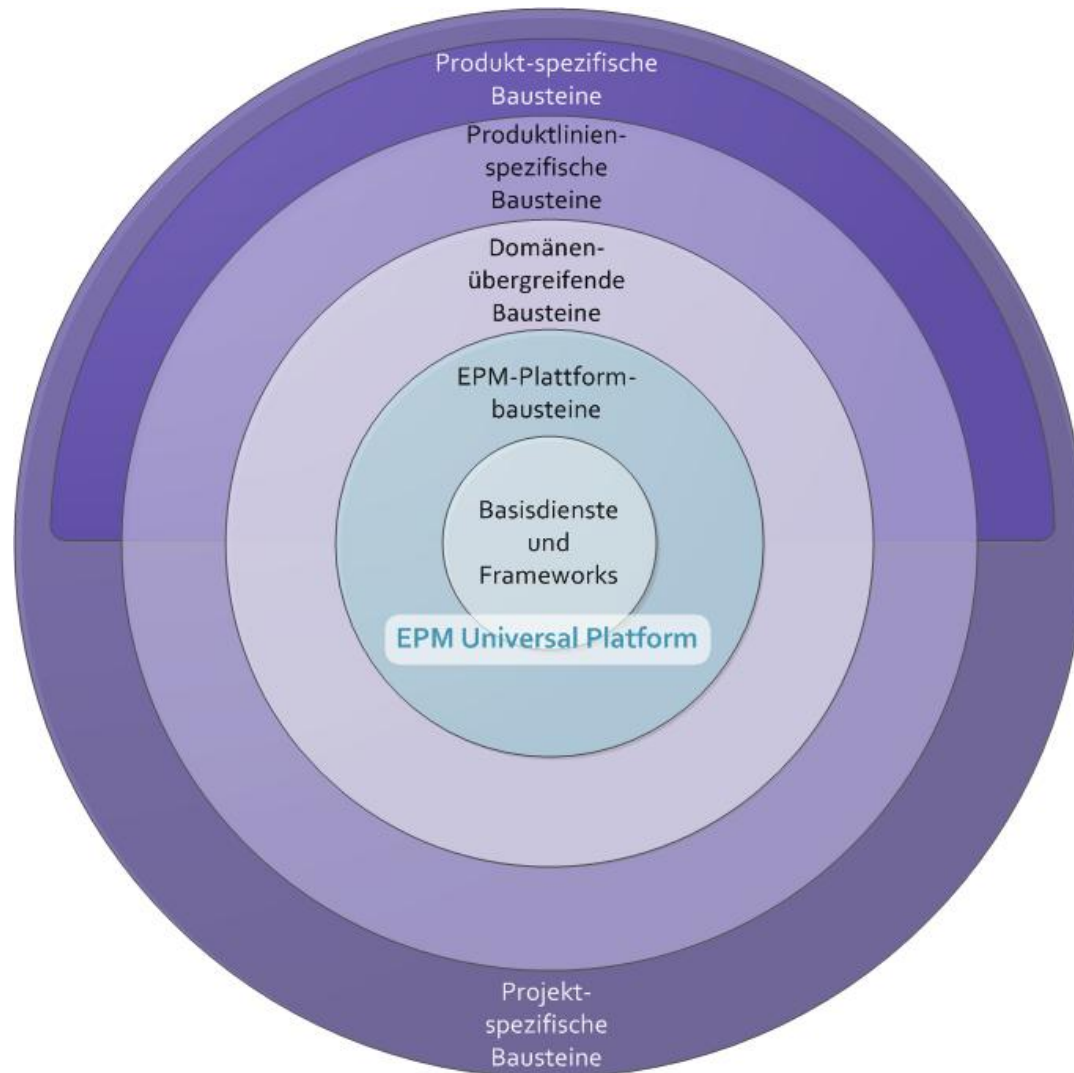
Gegenstand

- Portfolio von Softwareprodukten für das technisch-betriebliche Energie-Prozess-Management (EPM)
- Insgesamt lose, organisatorisch verteilte Produktlinie mit enger zusammenhängenden Teil-Produktlinien
- Zum Beispiel Anwendungen in den Bereichen Leittechnik für Energieversorgungsnetze, Kraftwerke und Windparks, Advanced Meter Management und virtuelle Kraftwerke.

Produktlinien-weite Artefakte

- EPM-Referenzarchitektur: Beschreibung und Abgrenzung der Zuständigkeiten von Bausteinen auf verschiedenen Ebenen
- EPM-Architekturstil: Richtlinien für die Strukturierung der EPM-Referenzarchitektur und für die Modellierung und Umsetzung von Softwarearchitekturen

Schichtenmodell der EPM-Referenzarchitektur



- **Motivation**
- **EPM-Produktlinie**
 - Überblick
 - EPM-Referenzarchitektur
 - EPM-Architekturstil
- **Architekturstandpunkte im EPM-Architekturstil**
- **Standardschnittstellen im Architekturstandpunkt Logische Softwarestruktur**
 - Schnittstellenarten
 - Nutzung von Standardschnittstellen zur Umsetzung nicht-funktionaler Variationspunkte
- **Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen**
- **Zusammenfassung & Ausblick**

Architekturstandpunkte im EPM-Architekturstil



Module

Physische
Softwarestruktur

Konfigurator/Konfiguration
Deploymenteinheiten

Physische
Auslieferung

Architektur-
standpunkte

Logische
Softwarestruktur

(Komponenten-)Schnittstellen
Komponenten

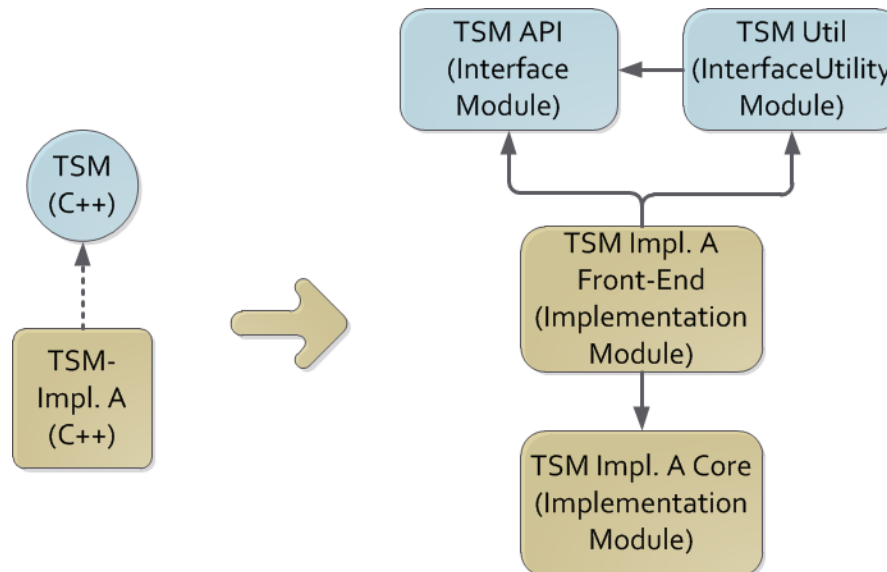
Logische
Verteilung

Kompositionskontext
Komponenten

Die Ausdrücke „Modul“ & „Komponente“ werden in vielfältigen Weisen verwendet, auch austauschbar und umgekehrt.

Im Rahmen des EPM-Architekturstils wird eine konsistente Sicht auf zwei unterschiedliche Konzepte mit diesen Namen definiert:

- Logische Komponenten werden durch Technologie-spezifische physische Module abgebildet



Modularisierung vs. Komponenten-Orientierung (2)

Gemeinsamer Gedanke ist Trennung von Schnittstellen und Implementierung

- Implementierungen ändern sich oft, Schnittstellen selten

Modularisierung ist das ältere Konzept (Parnas, 1972)

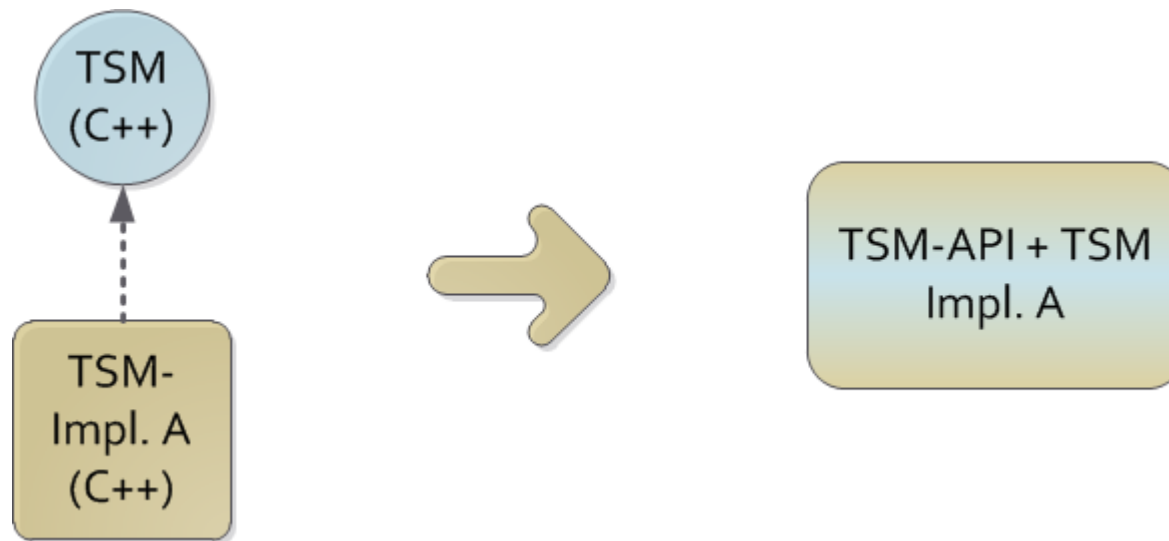
- Modul kapselt Entwurfsentscheidungen (insb. Abhängigkeiten) der Implementierung
 - Expliziter Export einer Schnittstelle
 - Expliziter Import von (exportierten) Schnittstellen anderer Module
- Ähnlich zu Klassen, aber grobgranularer
- Programmiersprachen, die Module als Sprachkonstrukt unterstützen:
 - Modula-2/3, Ada, Python, Ruby, ...
 - Leider aktuell nicht: C++, Java SE (aber OSGi), C#

(Statische) Komponenten-Orientierung geht über Modularisierung hinaus

- Komponenten-Orientierung setzt Modularisierung praktisch voraus, umgekehrt gilt dies nicht
- Explizite Trennung von Komponenten-Schnittstellen und Komponenten-Implementierungen
 - Unabhängiger Lebenszyklus & Versionierung
 - Komponenten-Schnittstellen bilden eigenständige Module

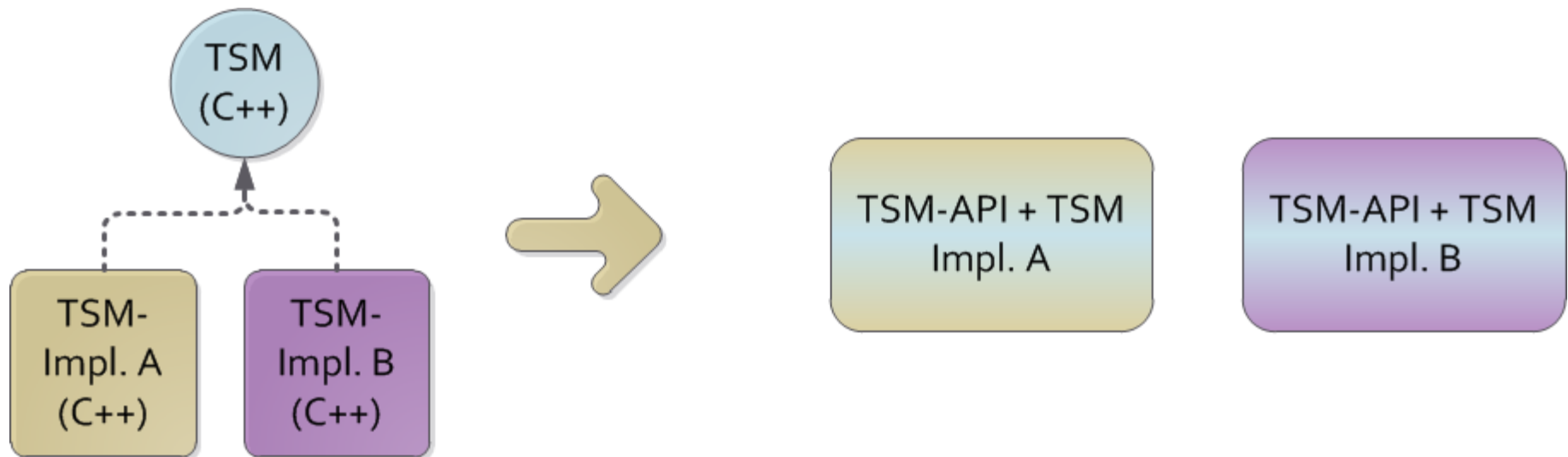
Komponenten-Plattformen bieten weitere dynamische Konzepte: Automatische Abhängigkeitsauflösung, automatische Verteilung von neuen Versionen, Laufzeitkonfiguration, etc.

Abbildung von Schnittstellen & Komponenten auf Module (1)



Modellierung von Komponenten und Schnittstellen,
modulare Umsetzung,
aber keine Komponenten-orientierte Abbildung!

Abbildung von Schnittstellen & Komponenten auf Module (2)



Modellierung von Komponenten und Schnittstellen,
modulare Umsetzung,
aber keine Komponenten-orientierte Abbildung!

Abbildung von Schnittstellen & Komponenten auf Module (3)

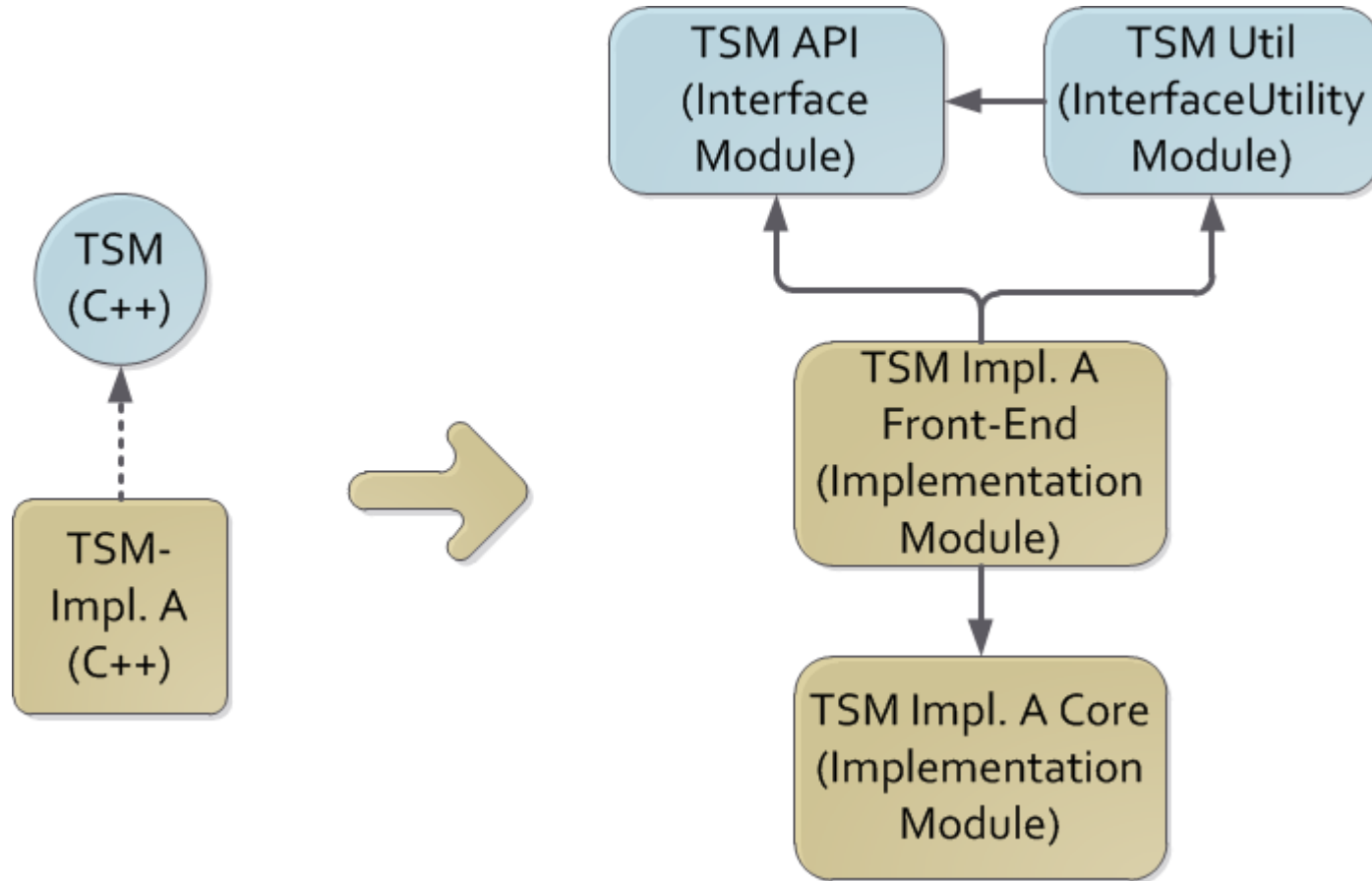
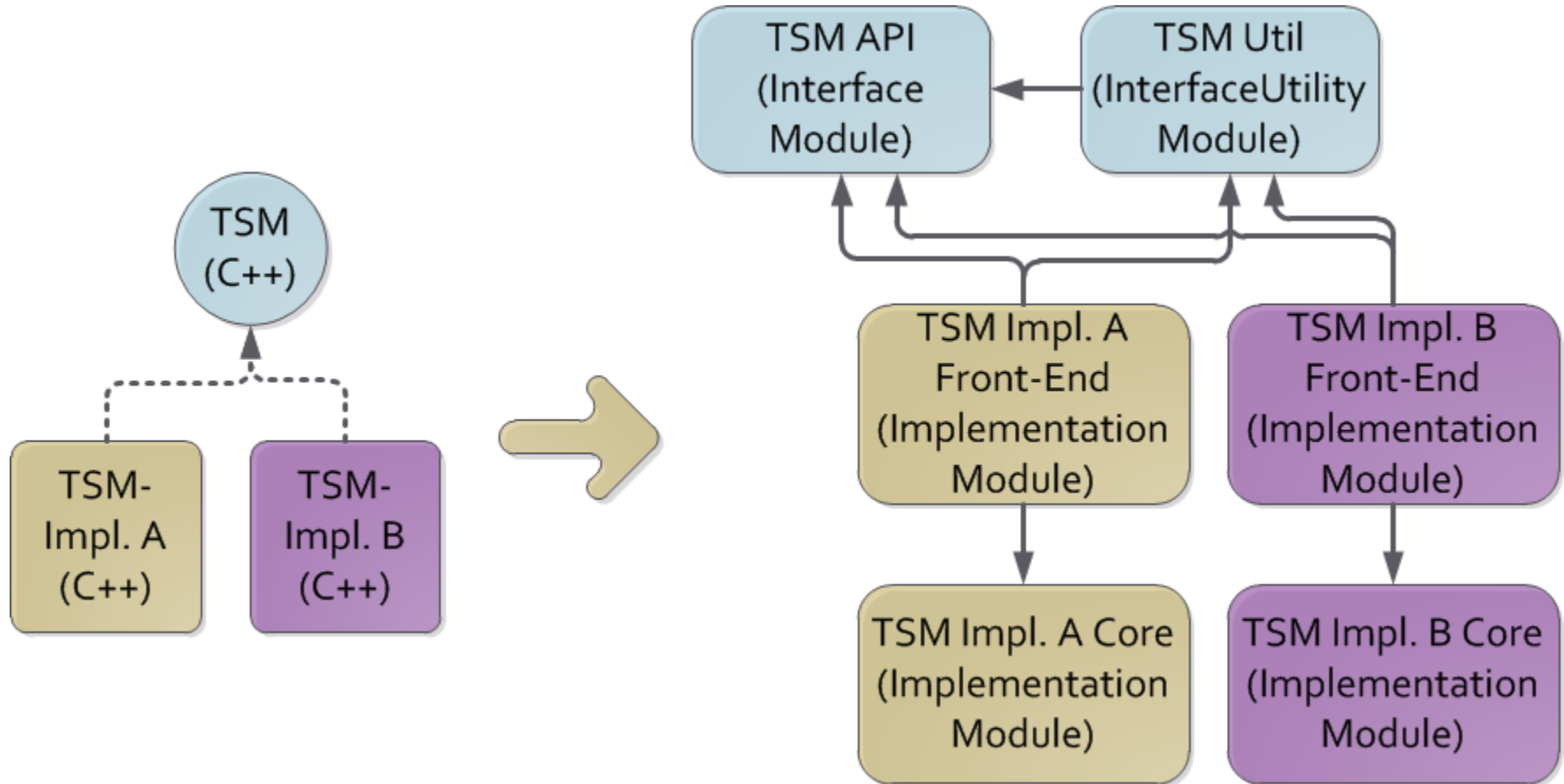


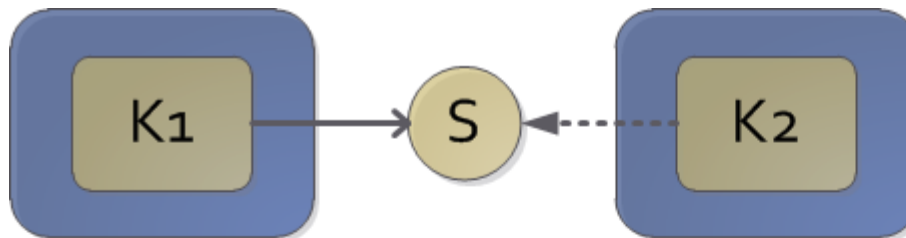
Abbildung von Schnittstellen & Komponenten auf Module (4)



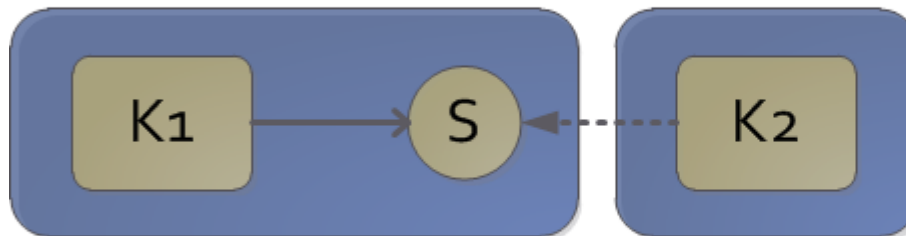
Wem „gehört“ die Schnittstelle? Wann darf sie geändert werden?

- **Motivation**
- **EPM-Produktlinie**
 - Überblick
 - EPM-Referenzarchitektur
 - EPM-Architekturstil
- **Architekturstandpunkte im EPM-Architekturstil**
- **Standardschnittstellen im Architekturstandpunkt Logische Softwarestruktur**
 - Schnittstellenarten
 - Nutzung von Standardschnittstellen zur Umsetzung nicht-funktionaler Variationspunkte
- **Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen**
- **Zusammenfassung & Ausblick**

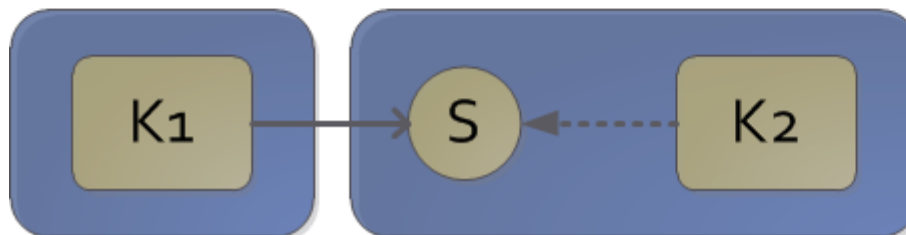
Besitzer von Schnittstellen



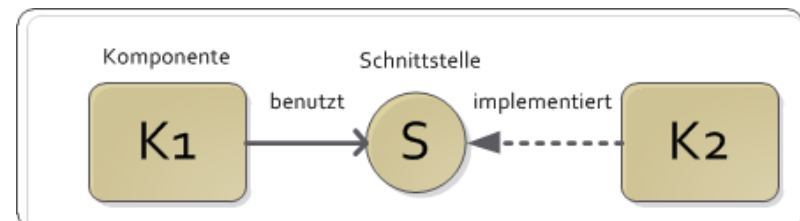
Viele Nutzer (Produkte),
Viele Implementierungen



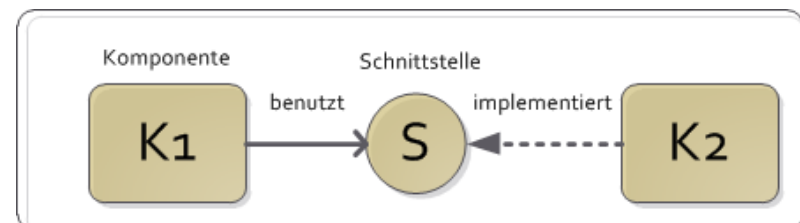
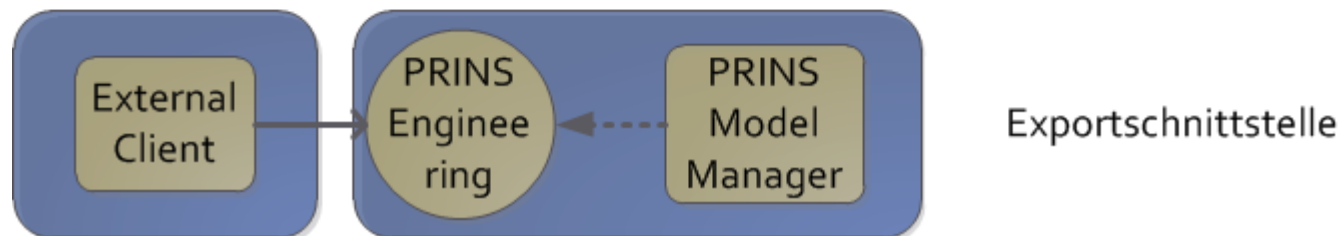
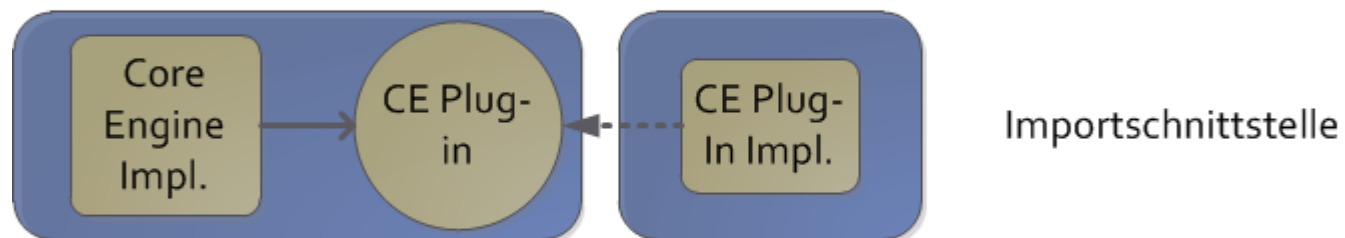
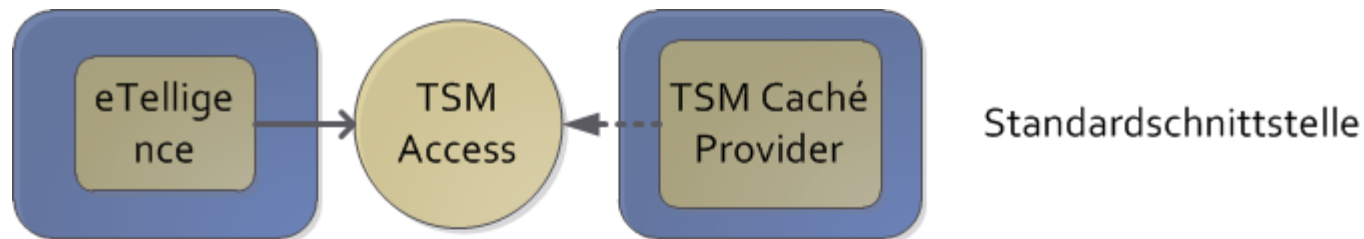
Ein Nutzer,
Viele Implementierungen



Viele Nutzer,
Eine Implementierung



Besitzer von Schnittstellen (Beispiele)



An Standardschnittstellen werden hohe Anforderungen gestellt:

- 1. Abgeschlossenheit**
- 2. Dokumentation**
- 3. Validierbarkeit**
- 4. Implementierbarkeit**
- 5. Verlässlichkeit**
- 6. Bekanntheit / Akzeptanz**

Nicht jede Schnittstelle, zu der mehr als eine Implementierung existiert, ist eine Standardschnittstelle.

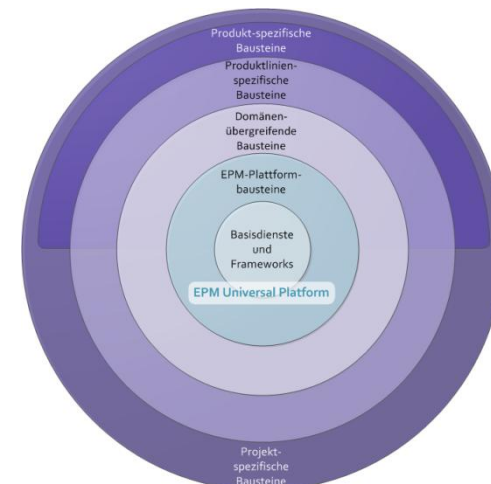
Eigenschaften, die bei Schnittstellen im Allgemeinen „verhandelbar“ sind, sind bei Standardschnittstellen zwingend.

Hohe anfängliche Kosten zur Reifung einer Standardschnittstelle, aber

- **Geringere Kosten je Nutzung durch Netzeffekte**
- **Austauschbarkeit**

Investition lohnt sich bei vielen Nutzern.

In die EPM Universal Platform werden gerade solche Bausteine eingeordnet, für die ein hinreichend großes Wiederverwendungspotenzial erwartet wird, so dass sich die Investition in die Entwicklung einer Standardschnittstelle rentiert.

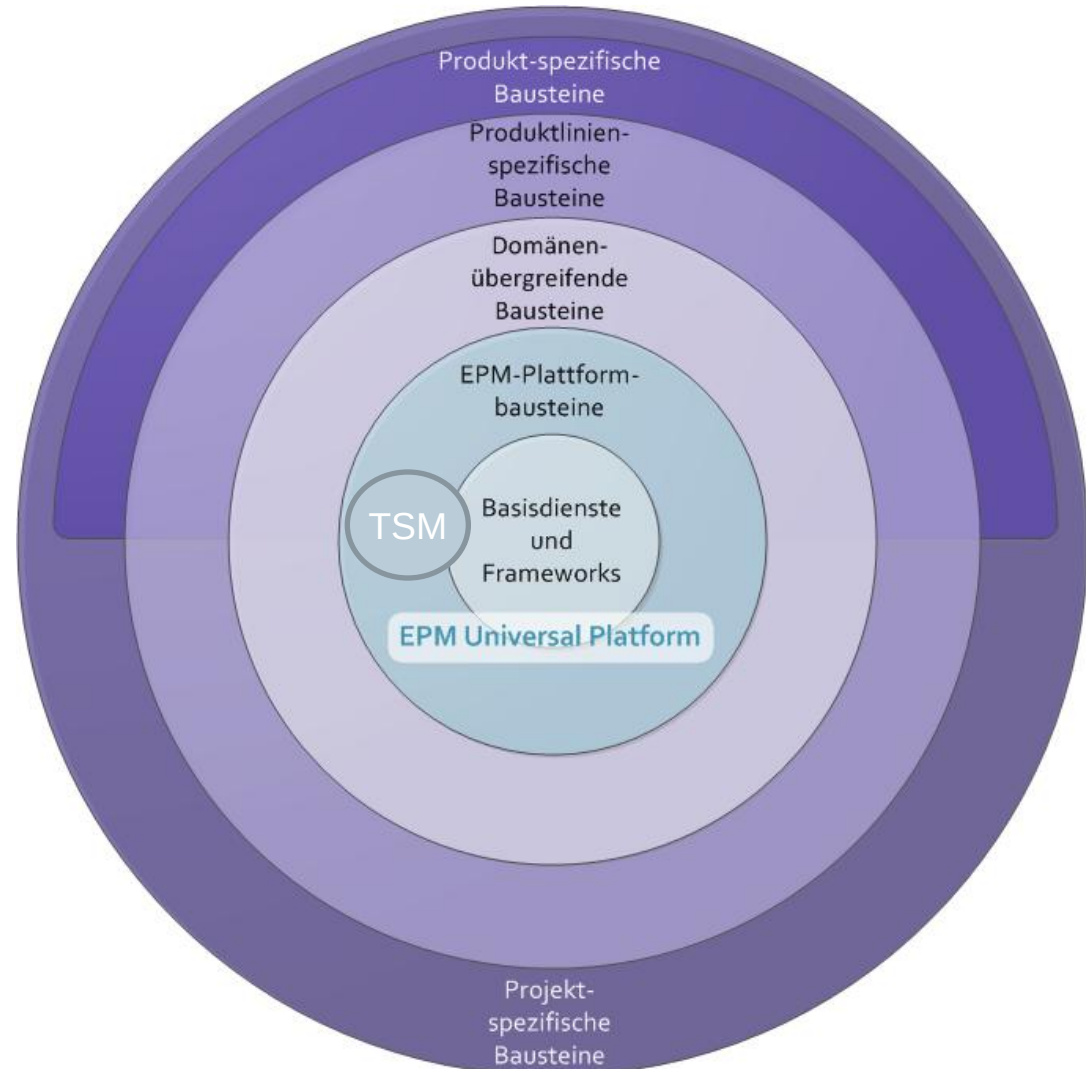


Bewertungsprozess für Plattformbaustein-Kandidaten

Bewertung:
Anzahl
potenzieller
Nutzer
etc.

?

Bewertung
positiv



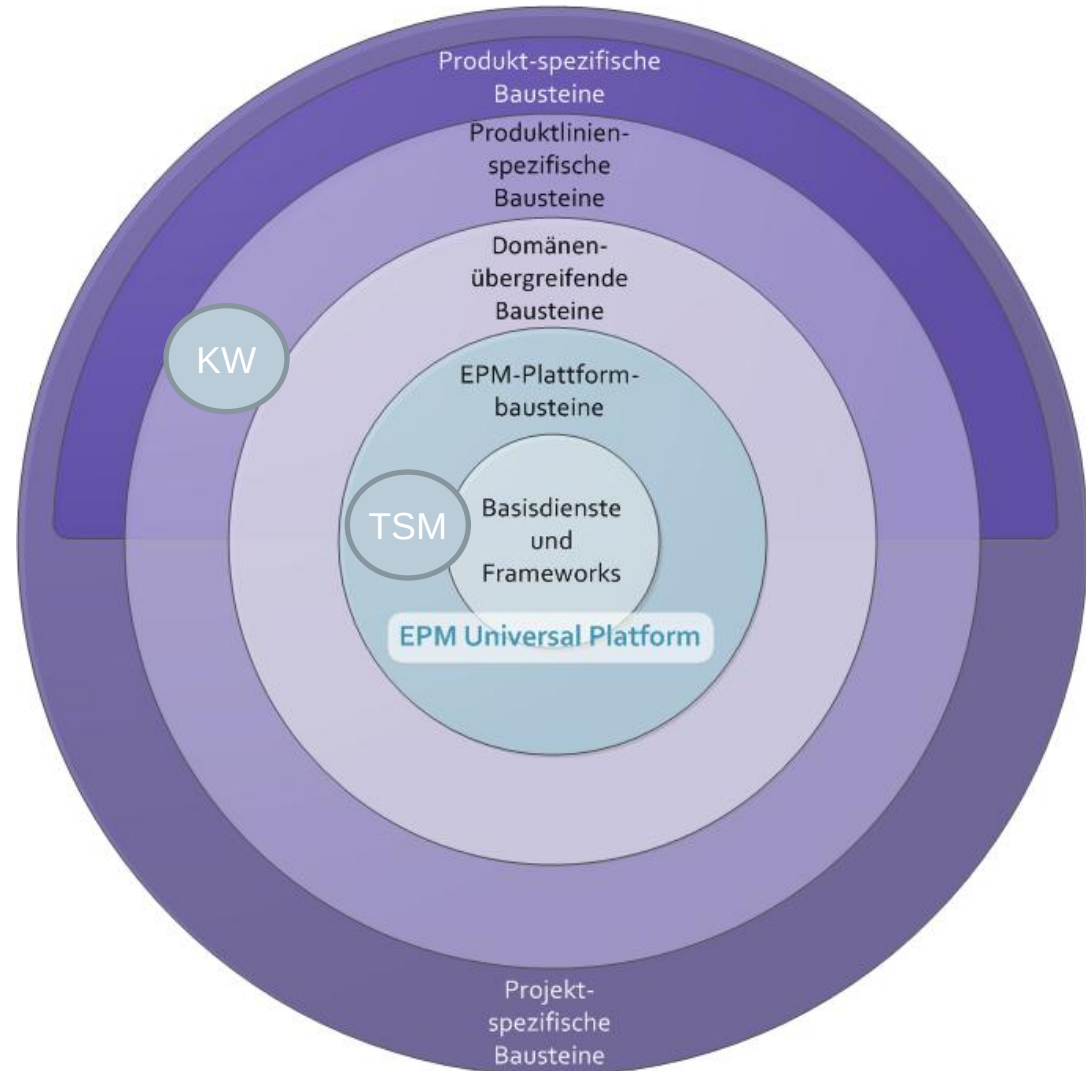
Bewertungsprozess für Plattformbaustein-Kandidaten

Bewertung:
Anzahl
potenzieller
Nutzer
etc.



?

Bewertung
negativ



- **Motivation**
- **EPM-Produktlinie**
 - Überblick
 - EPM-Referenzarchitektur
 - EPM-Architekturstil
- **Architekturstandpunkte im EPM-Architekturstil**
- **Standardschnittstellen im Architekturstandpunkt Logische Softwarestruktur**
 - Schnittstellenarten
 - Nutzung von Standardschnittstellen zur Umsetzung nicht-funktionaler Variationspunkte
- **Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen**
- **Zusammenfassung & Ausblick**

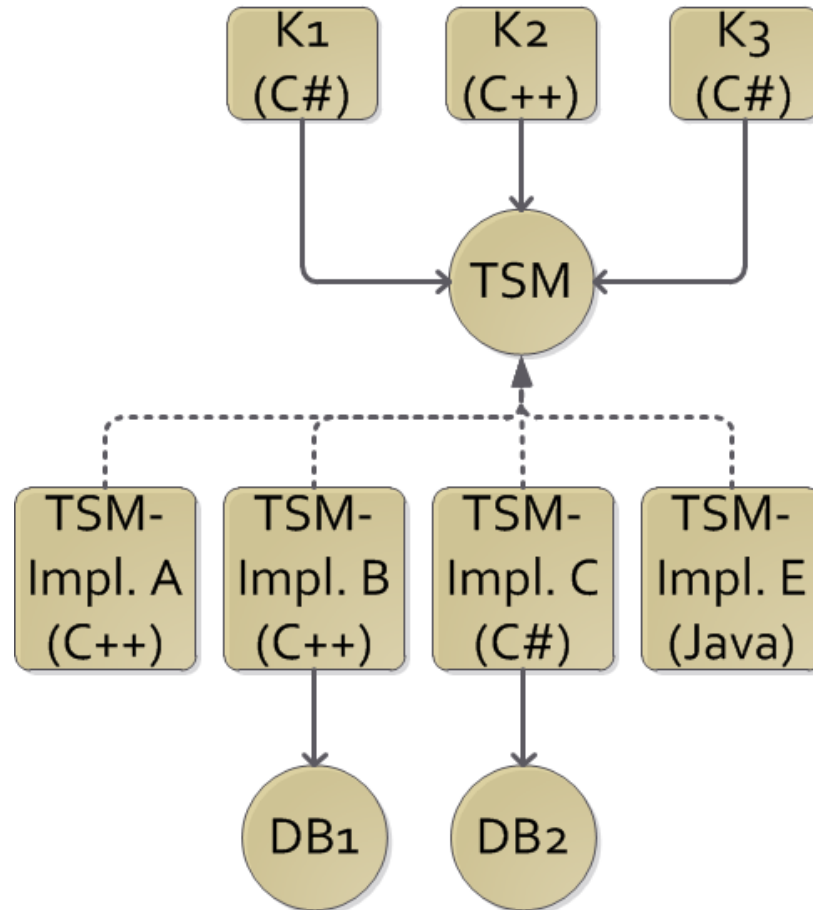
Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen



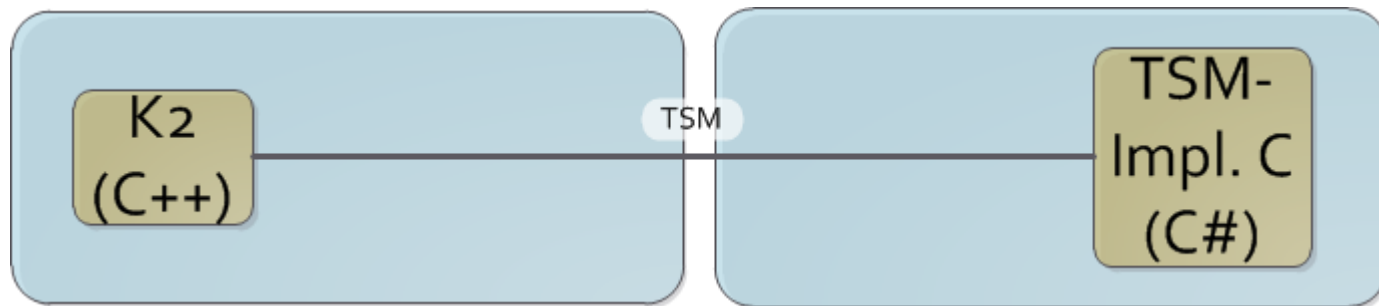
1. **Beispiel: Time Series Management (TSM)**
2. Parametrisierung von Standardschnittstellen-basierten Tests
3. Erweiterte Parametrisierung mit Test-Treibern
4. Strukturierung von Standardschnittstellen und zugehörigen Tests

Time Series Management: Implementierungen und Nutzer

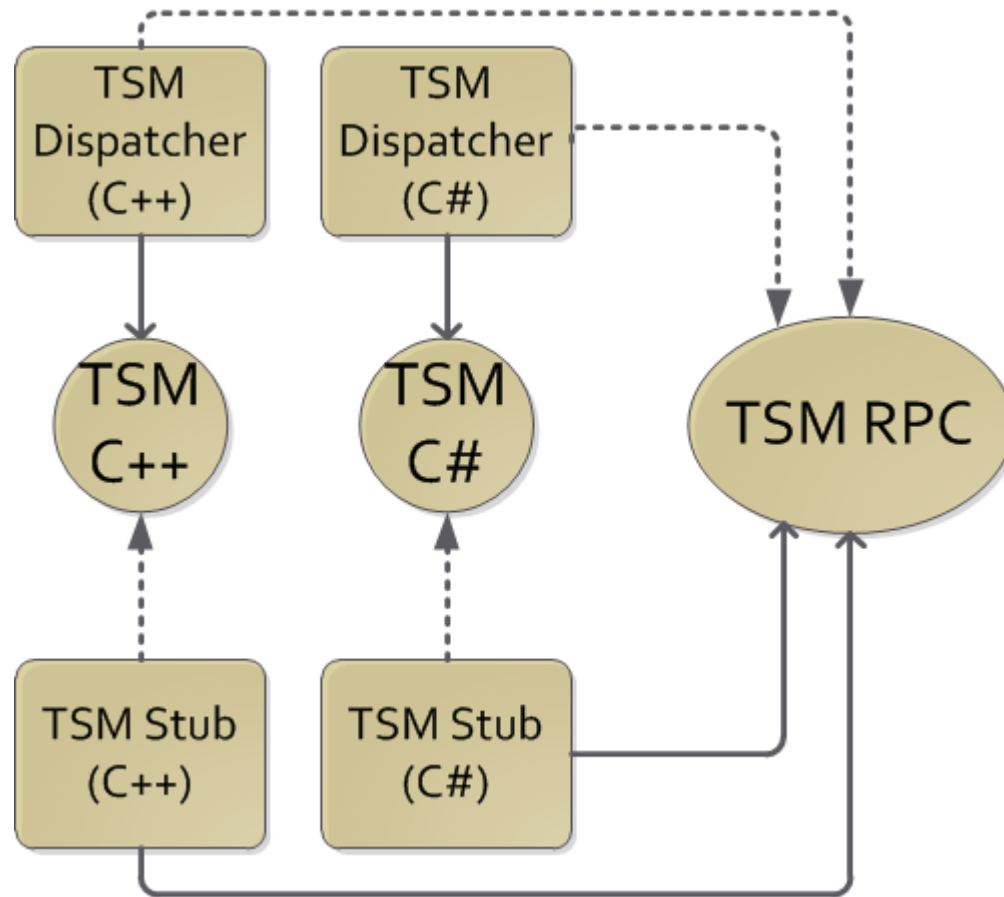
Logische Struktursicht



Technologie-übergreifende Nutzung von Standardschnittstellen: Logische Verteilungssicht

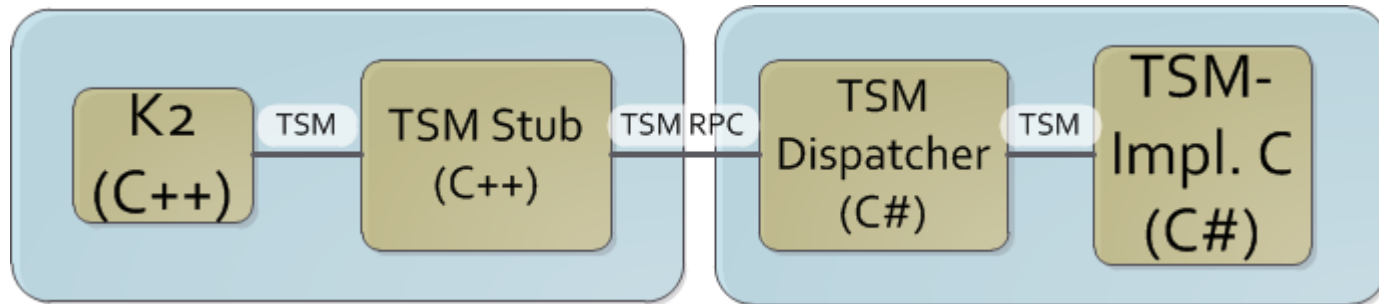


Technologie-übergreifende Nutzung von Standardschnittstellen am Beispiel von TSM: Logische Struktursicht



Technologie-übergreifende Nutzung von Standardschnittstellen:

Logische Verteilungssicht mit Stub&Dispatcher

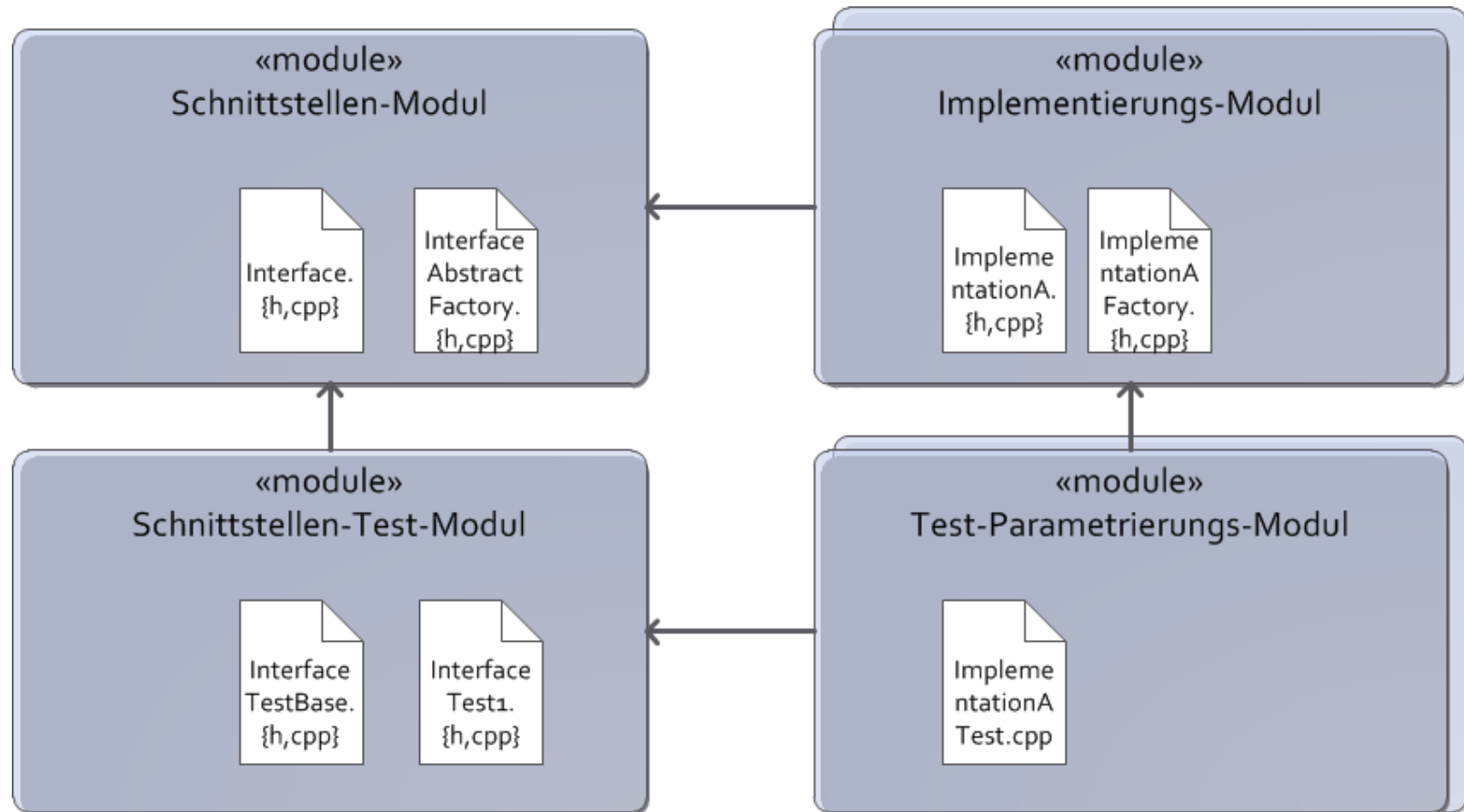


Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen

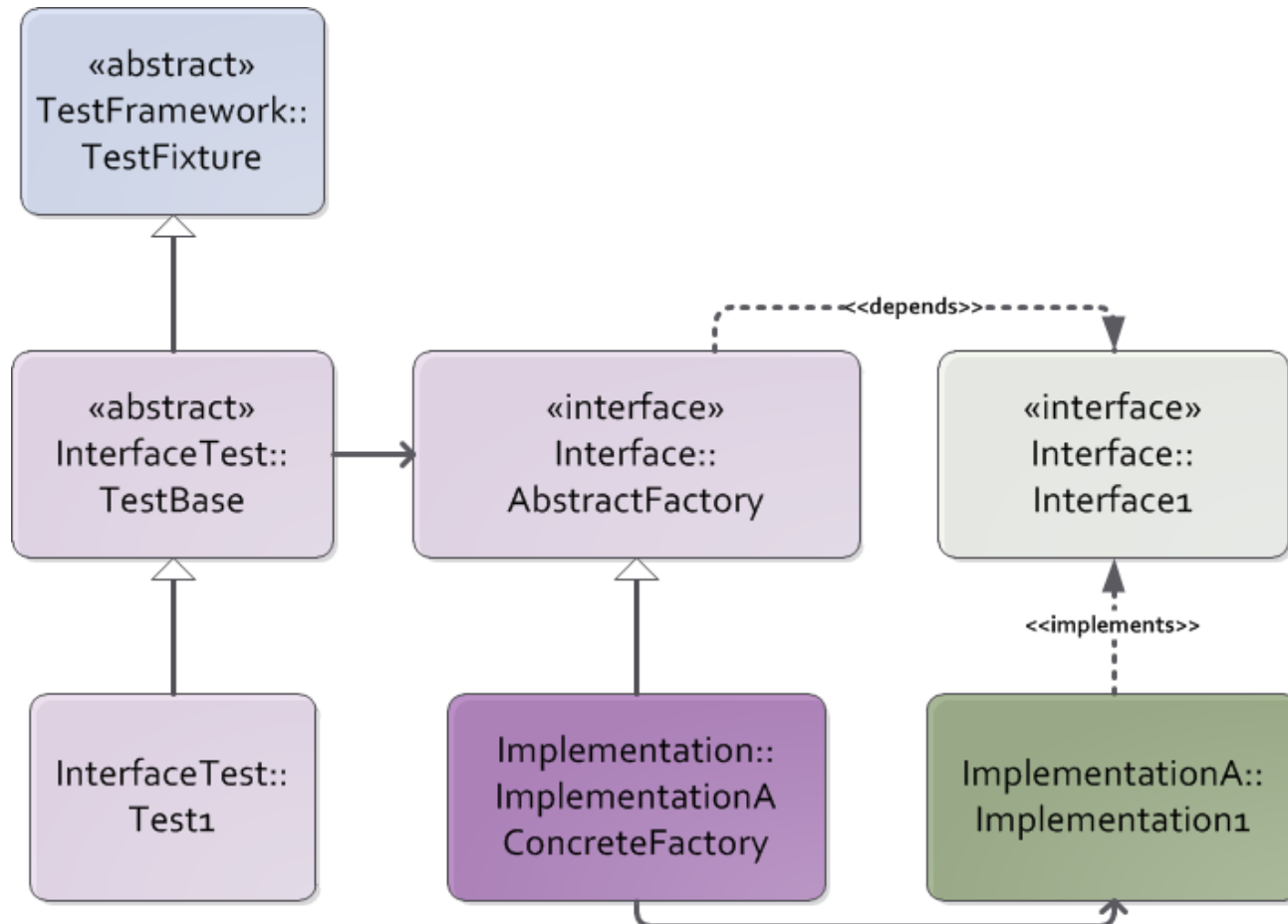


1. Beispiel: Time Series Management (TSM)
- 2. Parametrisierung von Standardschnittstellen-basierten Tests**
3. Erweiterte Parametrisierung mit Test-Treibern
4. Strukturierung von Standardschnittstellen und zugehörigen Tests

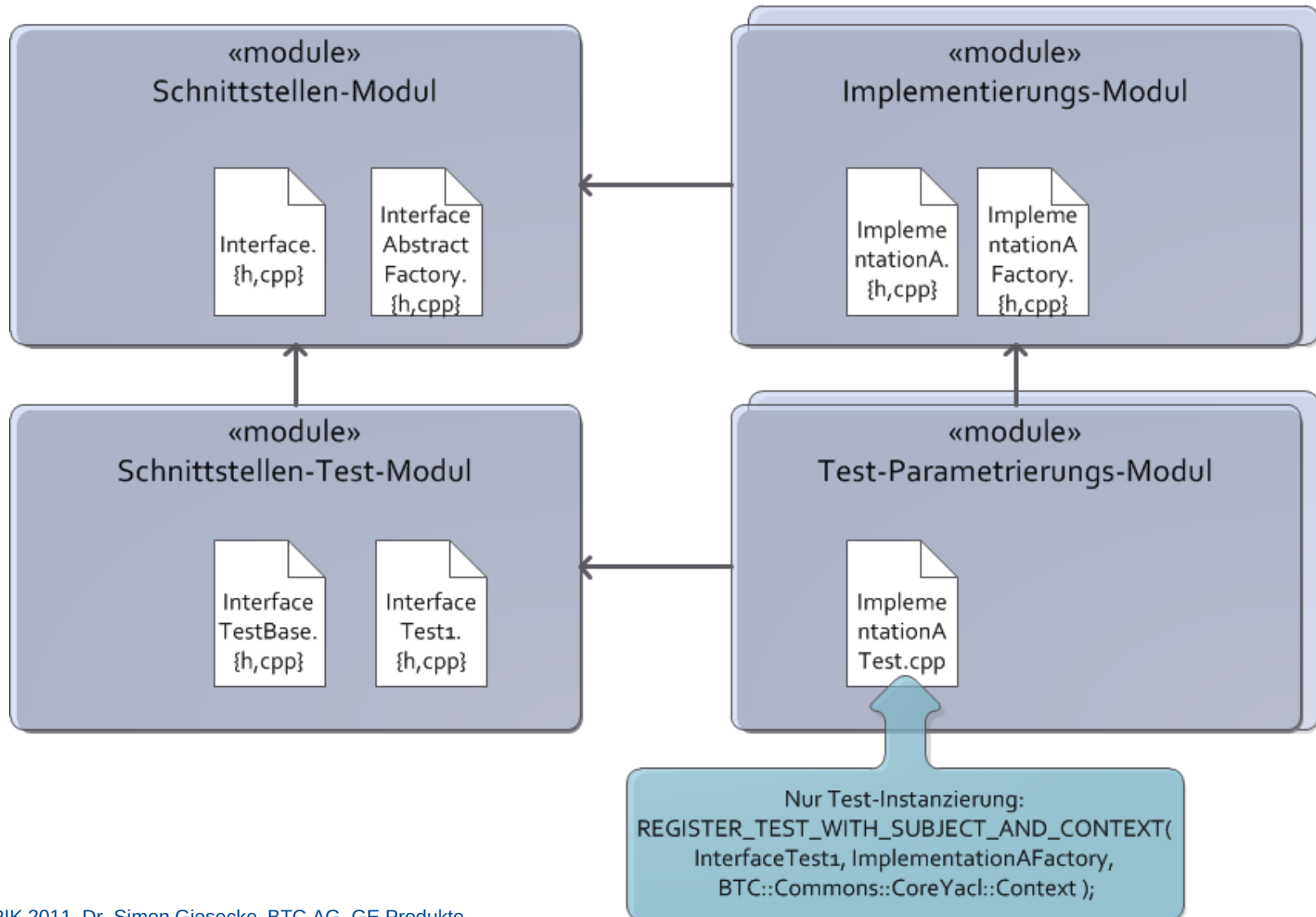
Parametrisierung von Standardschnittstellen-basierten Tests



Parametrisierung von Standardschnittstellen-basierten Tests



Parametrisierung von Standardschnittstellen-basierten Tests

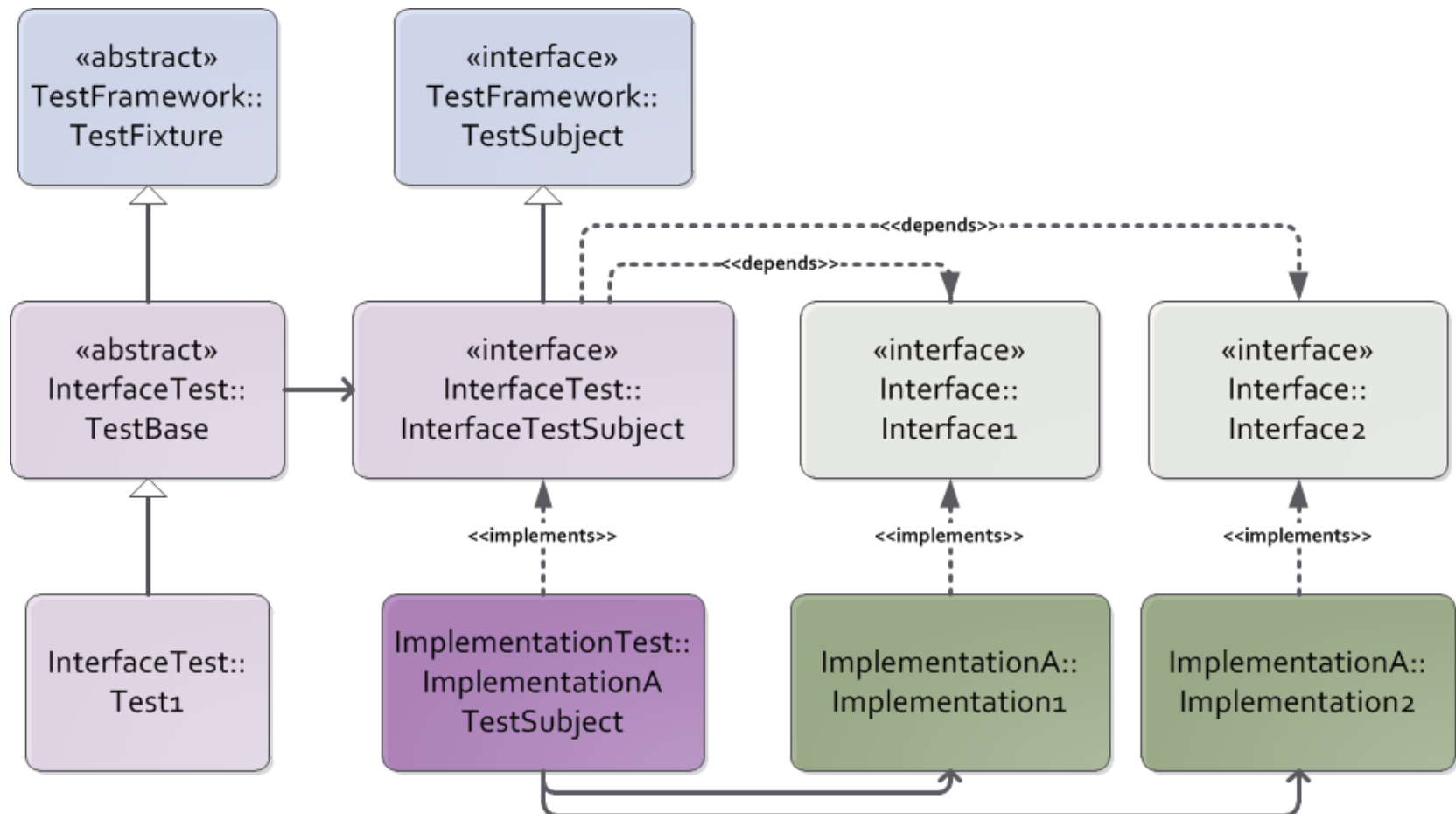


Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen



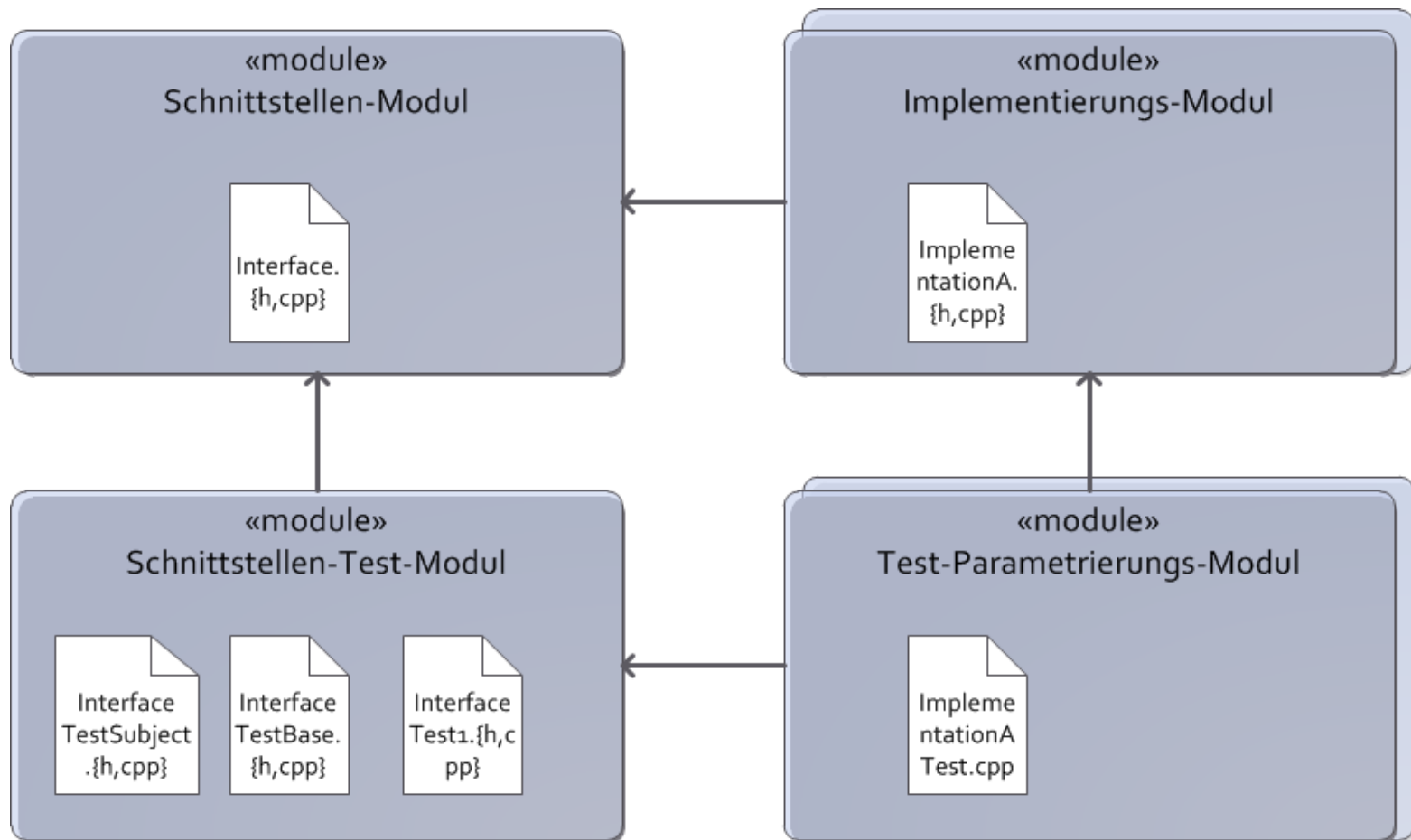
1. Beispiel: Time Series Management (TSM)
2. Parametrisierung von Standardschnittstellen-basierten Tests
- 3. Erweiterte Parametrisierung mit Test-Treibern**
4. Strukturierung von Standardschnittstellen und zugehörigen Tests

Parametrisierung von Tests: Allgemeines Muster (Klassenstruktur)



```
class ArchiveTestSubject :  
    public BTC::Test::TestFw::TestSubject,  
    public virtual api::ArchiveWriterFactory,  
    public virtual api::ArchiveReaderFactory {  
    public:  
        virtual const btc::coreengine::pv::DescriptorType &GetIntegerDescriptorType() const = 0;  
        virtual const btc::coreengine::pv::DescriptorTypeRegistry &GetDescriptorTypeRegistry()  
            const = 0;  
        virtual std::list<const btc::coreengine::pv::DescriptorType *> GetAllProcessVariableTypes()  
            const = 0;  
  
        virtual void SetUp() = 0;  
        virtual void TearDown() = 0;  
  
        virtual ~ArchiveTestSubject();  
};
```

Parametrisierung von Tests: Allgemeines Muster (Modulstruktur)



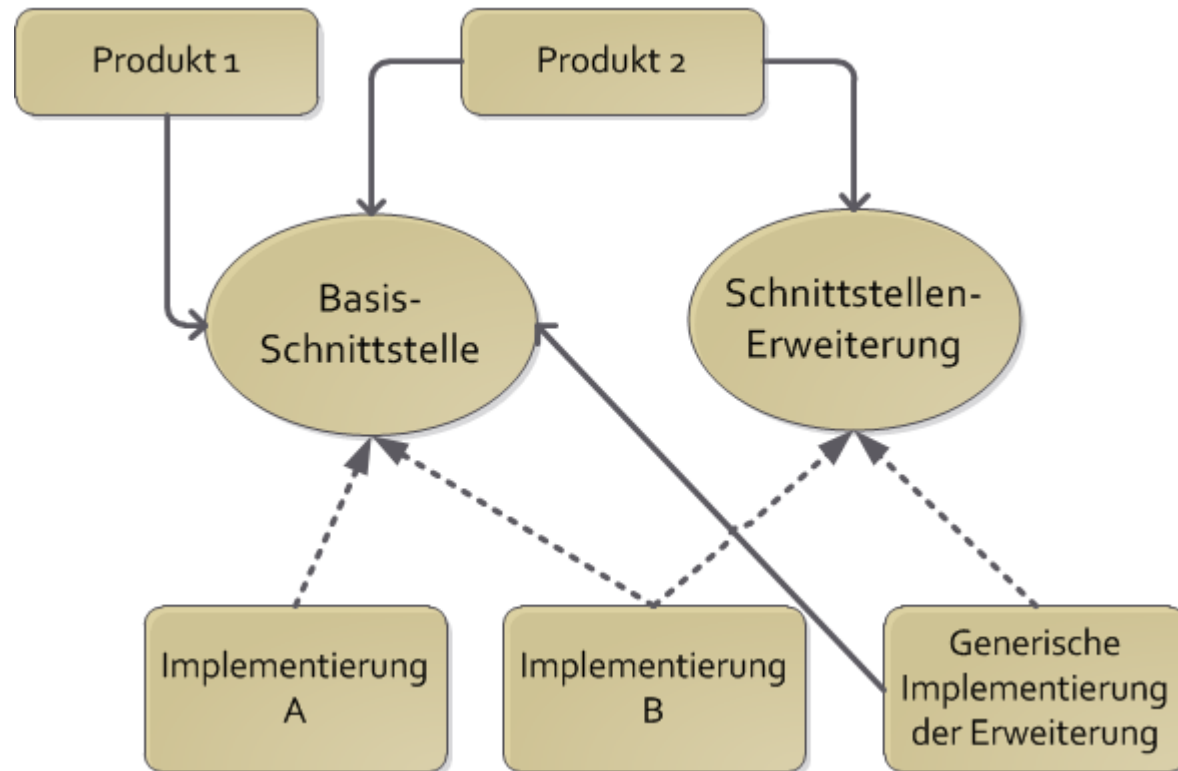


Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen

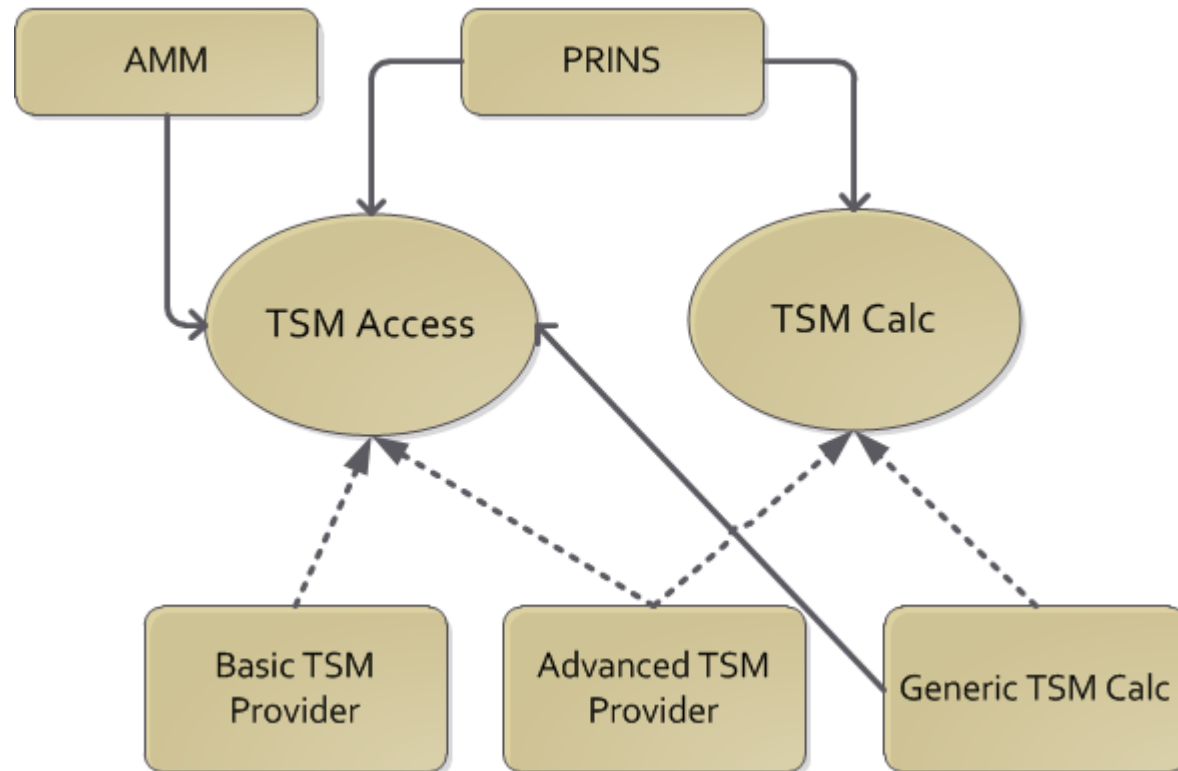


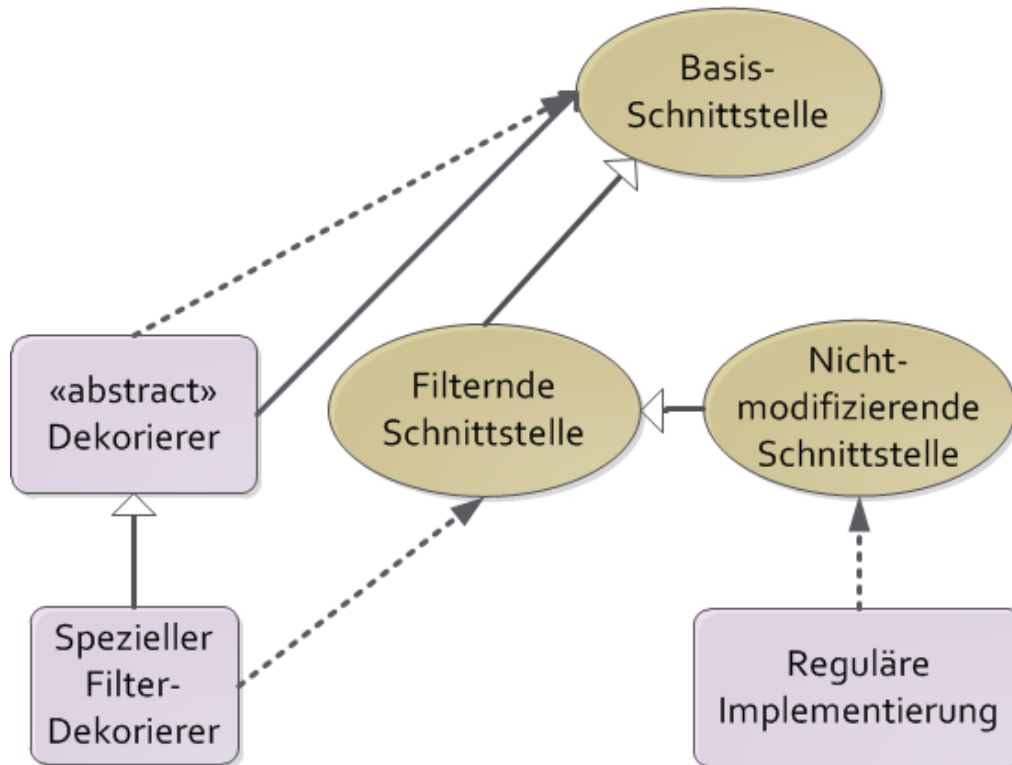
1. Beispiel: Time Series Management (TSM)
2. Parametrisierung von Standardschnittstellen-basierten Tests
3. Erweiterte Parametrisierung mit Test-Treibern
4. **Strukturierung von Standardschnittstellen und zugehörigen Tests**

Basis-Schnittstellen und Schnittstellen-Erweiterungen

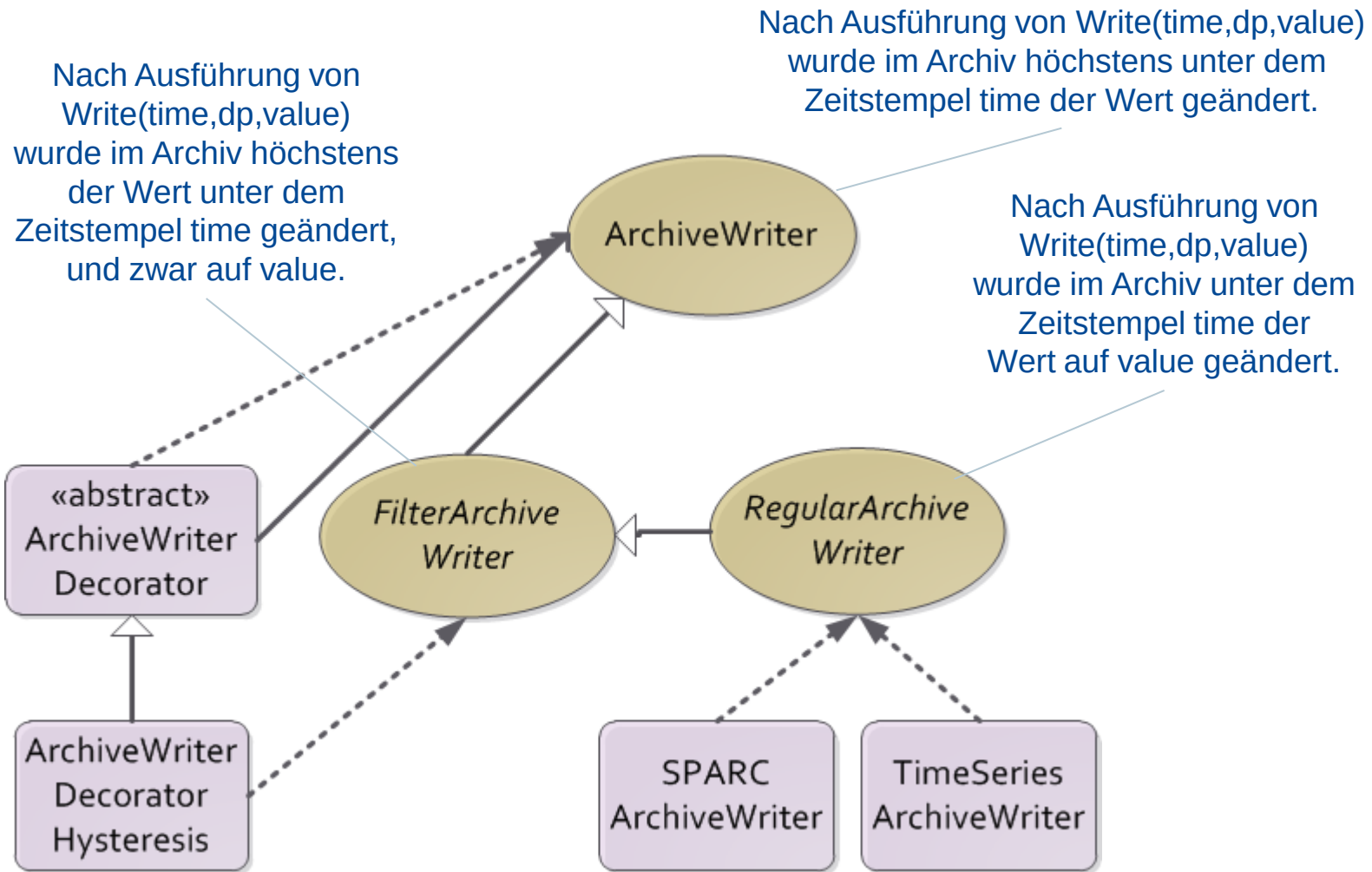


Basis-Schnittstellen und Schnittstellen-Erweiterungen: Beispiel





Schnittstellen-Hierarchie: Beispiel



- **Motivation**
- **EPM-Produktlinie**
 - Überblick
 - EPM-Referenzarchitektur
 - EPM-Architekturstil
- **Architekturstandpunkte im EPM-Architekturstil**
- **Standardschnittstellen im Architekturstandpunkt Logische Softwarestruktur**
 - Schnittstellenarten
 - Nutzung von Standardschnittstellen zur Umsetzung nicht-funktionaler Variationspunkte
- **Erfahrungen & Herausforderungen aus der Entwicklung von Standardschnittstellen**
- **Zusammenfassung & Ausblick**

Rolle von Standardschnittstellen in der EPM-Produktlinie

- Realisierung nicht-funktionaler Variationspunkte

Einordnung von Standardschnittstellen in den EPM-Architekturstil

- Modellierungselement der logischen Softwarestruktur

Anforderungen an Standardschnittstellen

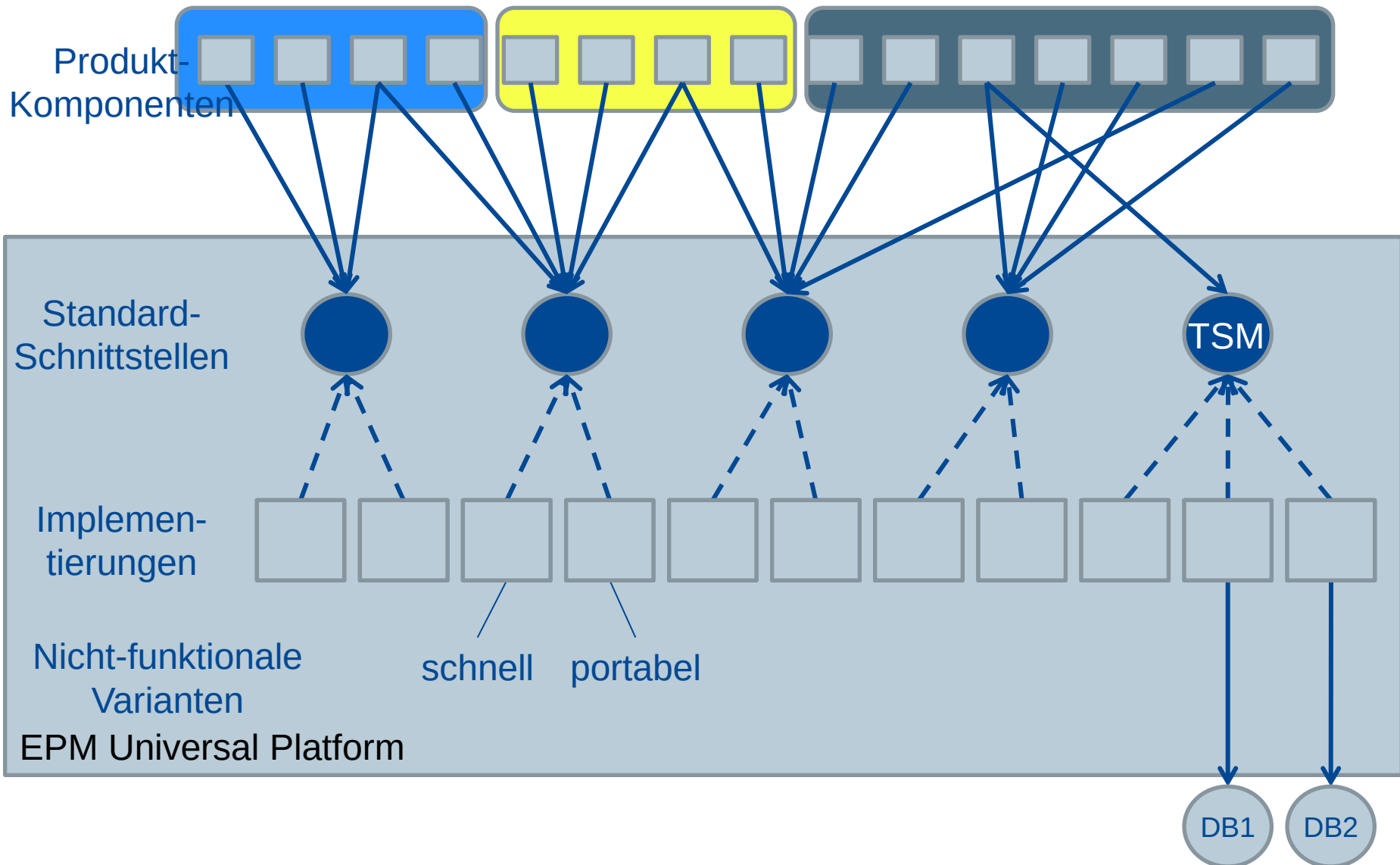
- Abgeschlossenheit, Dokumentation, Validierbarkeit, Implementierbarkeit, Verlässlichkeit, Bekanntheit / Akzeptanz

Entscheidungsprozess für die Klassifikation eines Bausteins als Standardschnittstelle

Beispiel, Erfahrungen und Herausforderungen

- Schnittstellen-basierte Tests erfordern sorgfältige Überlegungen
- Diese kosten aber nicht nur Aufwand, sondern erhöhen auch die Qualität und den Nutzen der Schnittstelle

Plattformbausteine der EPM Universal Platform



Neben Standardschnittstellen als nicht-funktionale Variationspunkte auch ein Standardkonzept für Plug-in-Schnittstellen (Import- oder Standardschnittstellen) für die Umsetzung funktionaler Variationspunkte

- Modellierung über Feature-Bäume
- Infrastruktur für Verwaltung von Plug-ins
- Wann sind Plug-in-Schnittstellen als Implementierungs-spezifische Importschnittstellen auszuführen, wann als Standard-Schnittstellen?

Codeabdeckung durch Schnittstellen-basierte Tests

- Mit einer Schnittstellen-basierten Testsuite wurden für eine Implementierung einer Schnittstelle eine Codeabdeckung von 99% erreicht.
- Andere, funktional äquivalente Implementierungen hatten mit derselben Testsuite nur Codeabdeckungen von 68% bzw. 40%
- Ursachen? Systematische Vermeidung möglich?

Die Standorte der BTC AG



Hauptsitz:

Escherweg 5
26121 Oldenburg
Fon: + 49 (0) 441/36 12-0
Fax: + 49 (0) 441/36 12-3999
E-Mail: office-ol@btc-ag.com
www.btc-ag.com



Luisenstraße 97
61348 Bad Homburg
Fon: +49 (0) 6172 68518-0
Fax: +49 (0) 6172 6722-50
E-Mail: office-ol@btc-ag.com

Kurfürstendamm 33
10719 Berlin
Fon: + 49 (0) 30 88096-5
Fax: + 49 (0) 30 88096-777
E-Mail: office-b@btc-ag.com

Mary-Somerville-Straße 3
28359 Bremen
Fon: +49 (0) 421 33039-0
Fax: +49 (0) 421 33039-399
E-Mail: office-hb@btc-ag.com

Wittekindstraße 32
44139 Dortmund
Fon: +49 (0) 231 981288-0
Fax: +49 (0) 231 981288-12
E-Mail: office-do@btc-ag.com

Bartholomäusweg 32
33334 Gütersloh
Fon: +49 (0) 5241 9463-0
Fax: +49 (0) 5241 9463-55
E-Mail: office-gt@btc-ag.com

Kehrwieder 9
20457 Hamburg
Fon: +49 (0) 40 210098-0
Fax: +49 (0) 40 210098-76
E-Mail: office-hh@btc-ag.com

Klostergasse 5
04109 Leipzig
Fon: +49 (0) 341 350558-0
Fax: +49 (0) 341 350558-59
E-Mail: office-l@btc-ag.com

Wilh.-Th.-Römheld-Str. 24
55130 Mainz
Fon: + 49 (0) 6131 88087-0
Fax: + 49 (0) 6131 88087-99
E-Mail: office-mz@btc-ag.com

Türkenstraße 55
80799 München
Fon: +49 (0) 89 3603539-0
Fax: +49 (0) 89 3603539-59
E-Mail: office-m@btc-ag.com

An der Alten Ziegelei 1
48157 Münster
Fon: +49 (0) 251 14132-0
Fax: +49 (0) 251 14132-11
E-Mail: office-ms@btc-ag.com

Konrad-Zuse-Straße 3
74172 Neckarsulm
Fon: +49 (0) 7132 380-0
Fax: +49 (0) 7132 380-29
E-Mail: office-nsu@btc-ag.com

Çayırıolu 1, Partaş Center Kat: 11-12
İçerenköy, 34752 İstanbul
Türkei
Fon: +90 (216) 5754590
Fax: +90 (216) 5754595
E-Mail: office-ist@btc-ag.com

ul. Mała Garbary 9
61-756 Poznań
Polen
Fon: +48 (0) 61 8560970
Fax: +48 (0) 61 8501870
E-Mail: biuro-poz@btc-ag.com

Hasebe Build.11F,
4-22-3 Sendagi, Bunkyo-Ku,
113-0022 Tokyo
Japan
Fon: +81 (3) 5832 7020
Fax: +81 (3) 5832 7021
Email: Info.OSCJapan@btc-es.de

Bäulerstraße 20
CH-8152 Glattbrugg
Schweiz
Fon: +41 (0) 44 874 3000
Fax: +41 (0) 44 874 3010
E-Mail: office-zh@btc-ag.com