LARA
A PHP Enterprise Application Framework

Seminar zum Schwerpunkt Systemtechnik

von
Kevin Wennemuth

4. September 2007

.: Überblick :.

- PHP Frameworks
- Architekturen und Konzepte
- PHP in der Neuzeit
- 30 Minuten Vortrag
- 15 Minuten Diskussion

.: Inhalt dieses Vortrages :.

.: Definition :.

”Ein Framework ist ein wiederverwendbarer Entwurf,
der durch eine Menge von abstrakten Klassen,
sowie dem Zusammenspiel ihrer Instanzen beschrieben wird.“

.: Was ist ein Framework? :.

- Reuse of analysis
  - Analyse der Anwendungsdomäne

.: Was ist ein Framework? :.

- Reuse of analysis
  - Analyse der Anwendungsdomäne
- Hot spots und hooks

.: Was ist ein Framework? :.

- Reuse of analysis
  - Analyse der Anwendungsdomäne
- Hot spots und hooks
  - Angriffspunkte für Anwender und Entwickler

.: Was ist ein Framework? :.

- Reuse of analysis
  - Analyse der Anwendungsdomäne
- Hot spots und hooks
  - Angriffspunkte für Anwender und Entwickler
- Inversion of control

.: Was ist ein Framework? :.

- Reuse of analysis
  - Analyse der Anwendungsdomäne
- Hot spots und hooks
  - Angriffspunkte für Anwender und Entwickler
- Inversion of control
  - Anwendungsfluss liegt bei dem Framework

.: Was ist ein Framework? :.

- Reuse of analysis
  - Analyse der Anwendungsdomäne
- Hot spots und hooks
  - Angriffspunkte für Anwender und Entwickler
- Inversion of control
  - Anwendungsfluss liegt bei dem Framework

.: Java :.

- typorientierte Programmiersprache
- strikte Objektparadigmen
- geringe Flexibilität
- hoher Wartungsaufwand

.: Java :.

- typorientierte Programmiersprache
- strikte Objektparadigmen
- geringe Flexibilität
- hoher Wartungsaufwand

.: PHP :.

- typfreie Programmiersprache
- hohe Entwicklungsgeschwindigkeit
- typfreie Programmiersprache
- keine strikten Objekthierarchien

.: PHP :.

- typfreie Programmiersprache
- hohe Entwicklungsgeschwindigkeit
- typfreie Programmiersprache
- keine strikten Objekthierarchien

.: Haben oder nicht haben... :.

- Datenbankabstraktion und Pooling
- Scaffolding und O/R-Mapping
- MVC2 Design-Pattern und Objektorientierung
- Templating und Caching

.: Haben oder nicht haben... :.

- Datenbankabstraktion und Pooling
- Scaffolding und O/R-Mapping
- MVC2 Design-Pattern und Objektorientierung
- Templating und Caching
- Anbindung an andere Systeme (SOAP, XML-RPC, ALE)
- Dokumentation

.: Haben oder nicht haben. . . :.

- Datenbankabstraktion und Pooling
- Scaffolding und O/R-Mapping
- MVC2 Design-Pattern und Objektorientierung
- Templating und Caching
- Anbindung an andere Systeme (SOAP, XML-RPC, ALE)
- Dokumentation

.: PEAR - PHP Extension and Application Repository :.

- gute Funktionalitäten
- exotische Aufgaben bereits gelöst
- keine Objektorientierung
- Dokumentation nicht vorhanden

.: PEAR - PHP Extension and Application Repository :.

- gute Funktionalitäten
- exotische Aufgaben bereits gelöst
- keine Objektorientierung
- Dokumentation nicht vorhanden

.: symfony :.

- großer Funktionsumfang
- sehr gutes Konzept
- kein Templating
- sehr langsamer Kern

.: symfony :.

- großer Funktionsumfang
- sehr gutes Konzept
- kein Templating
- sehr langsamer Kern

.: CodeIgniter :.

- großer Funktionsumfang
- sehr gute Dokumentation
- kein Design-Pattern
- Beta-Stadium

.: CodeIgniter :.

- großer Funktionsumfang
- sehr gute Dokumentation
- kein Design-Pattern
- Beta-Stadium

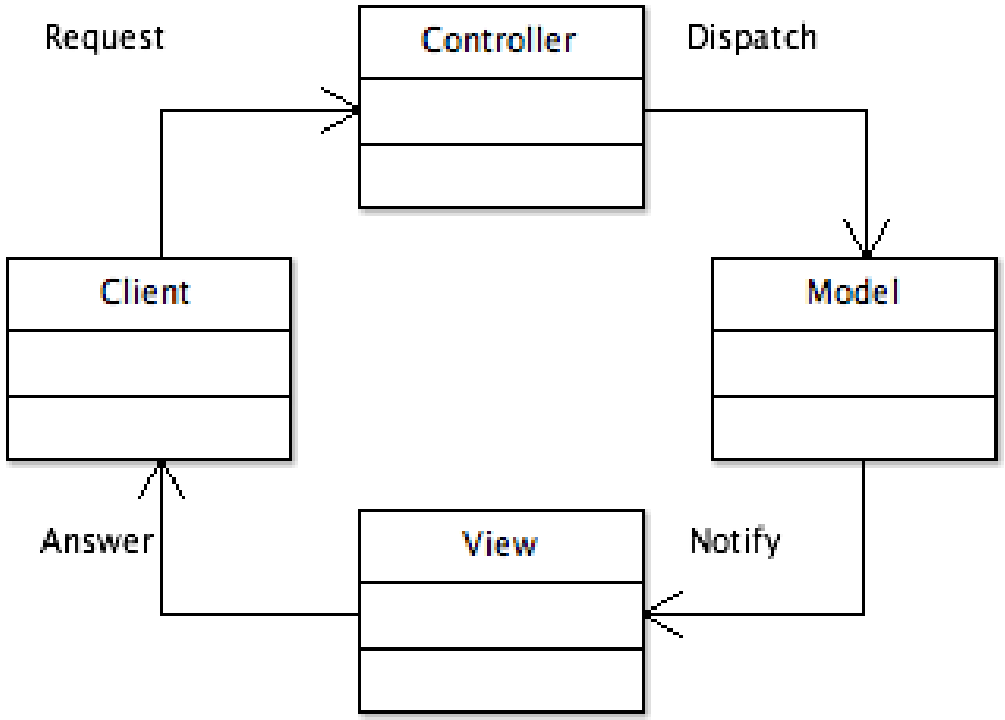
.: Warum Design-Pattern? :.

- Strukturierung der Architektur
- hohe Wiederverwendbarkeit
- Trennung von Kontrolle, Ansicht und Datenmodell
- Reduzierung der Komplexität

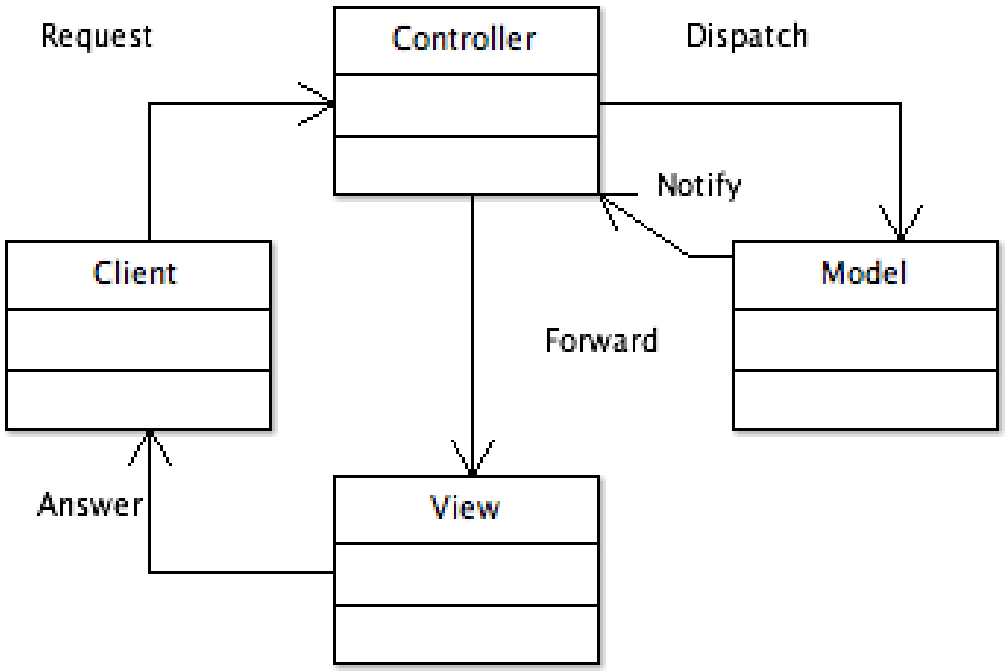
.: Warum Design-Pattern? :.

- Strukturierung der Architektur
- hohe Wiederverwendbarkeit
- Trennung von Kontrolle, Ansicht und Datenmodell
- Reduzierung der Komplexität

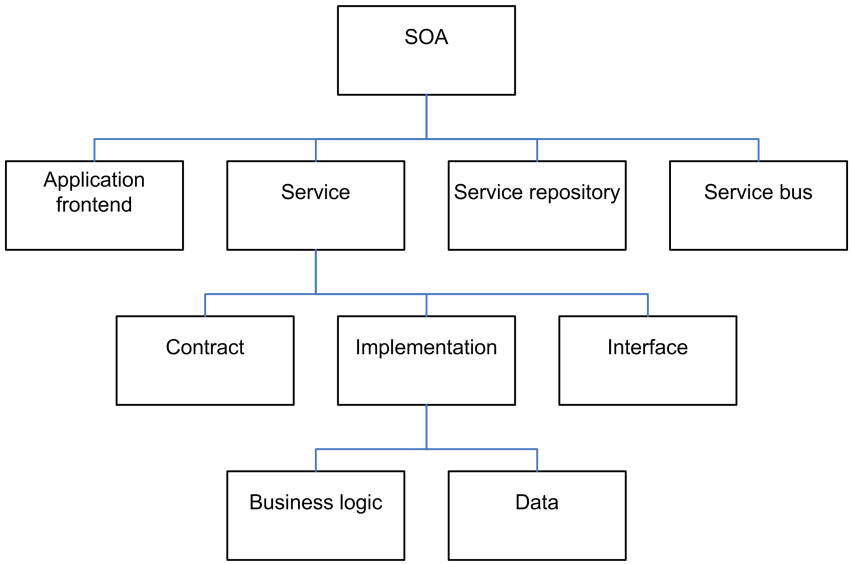
.: Model-View-Controller M. 1 :.



.: Model-View-Controller M. 2 :.

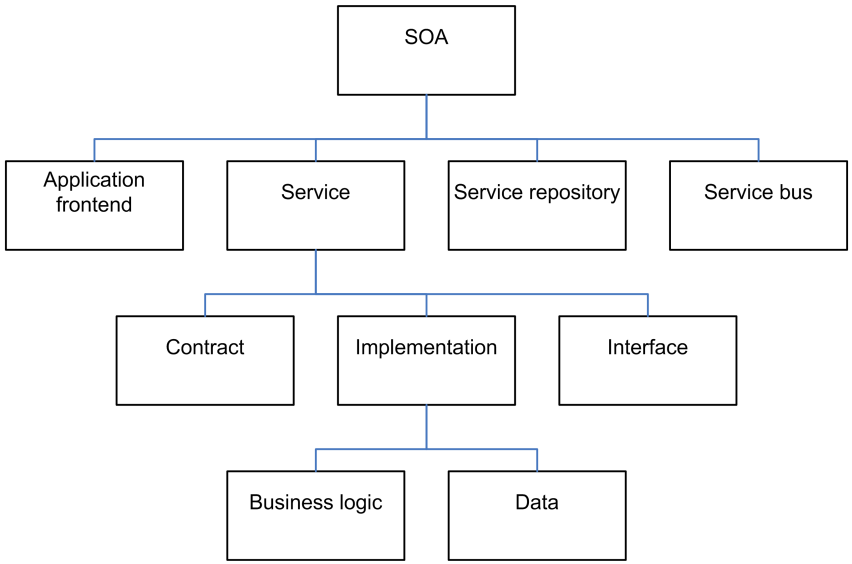


.: Serviceorientierte Architektur (SOA) :.



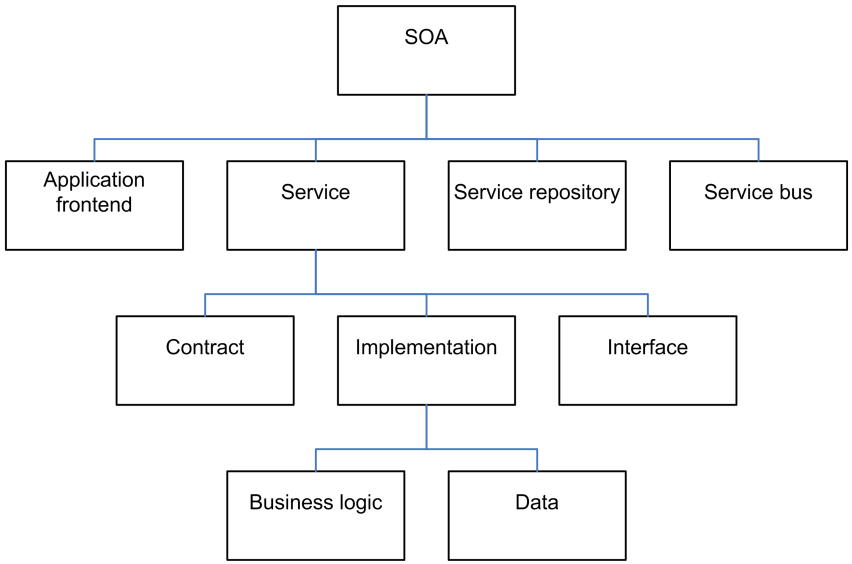
- MVC auf Konzeptebene
- geschäftsprozessorientiert

.: Serviceorientierte Architektur (SOA) :.



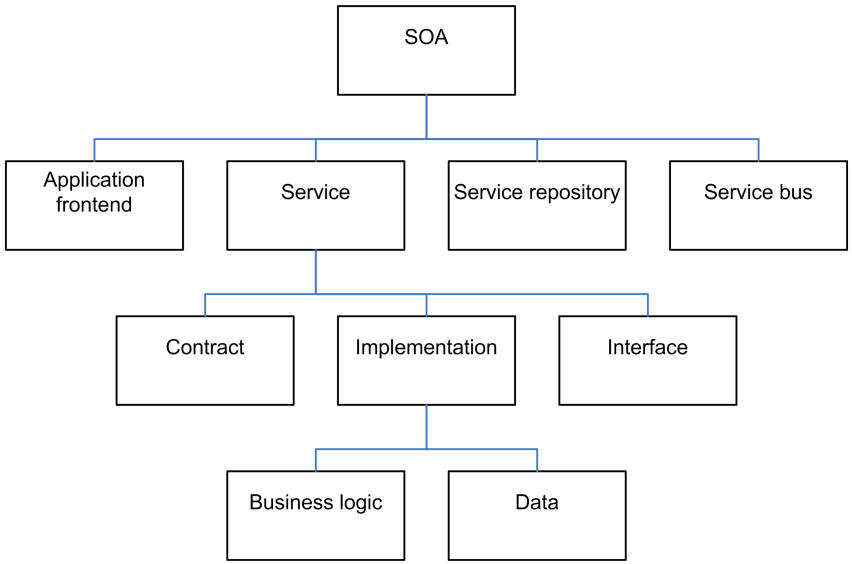
- MVC auf Konzeptebene
- geschäftsprozessorientiert
- atomare Prozesse

.: Serviceorientierte Architektur (SOA) :.



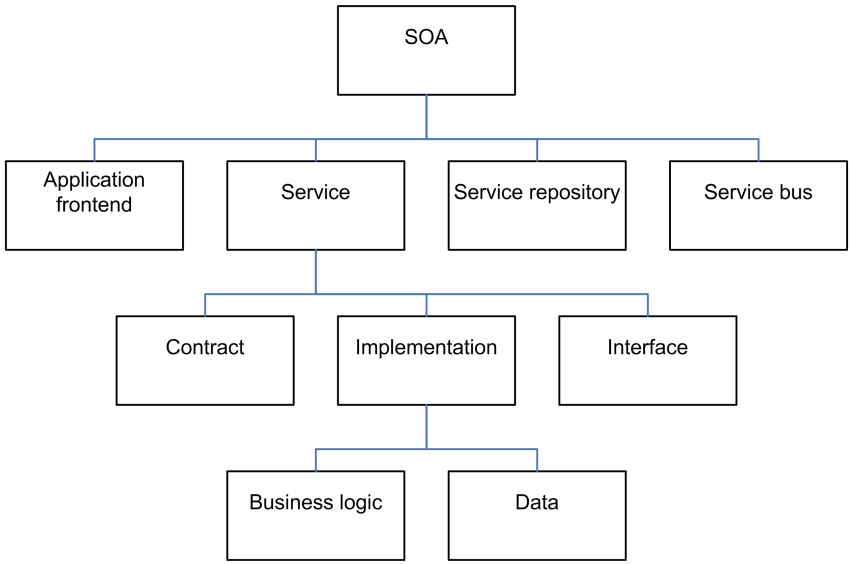
- MVC auf Konzeptebene
- geschäftsprozessorientiert
- atomare Prozesse
- Services

.: Serviceorientierte Architektur (SOA) :.



- MVC auf Konzeptebene
- geschäftsprozessorientiert
- atomare Prozesse
- Services
- Komposition

.: Serviceorientierte Architektur (SOA) :.



- MVC auf Konzeptebene
- geschäftsprozessorientiert
- atomare Prozesse
- Services
- Komposition

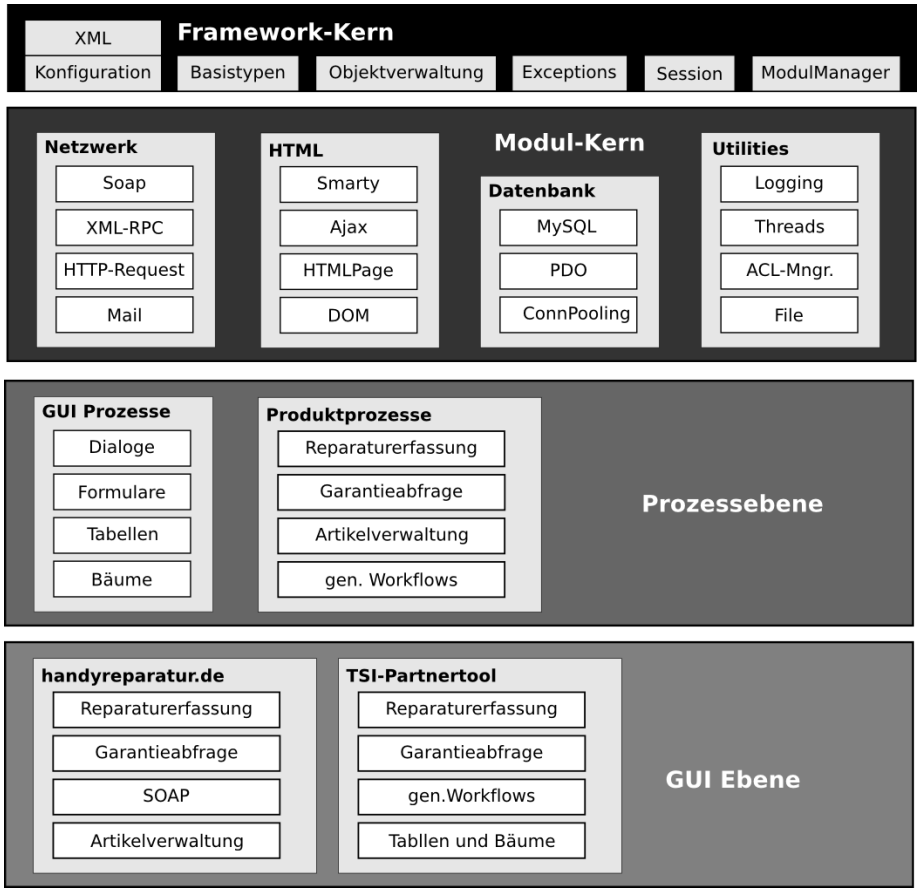
.: LARA - Logistic And Repair Application :.

- Refactoring alter Komponenten
- Repair Management Markt
- generisches Prozess-Framework
- SOA, ALE

.: LARA - Logistic And Repair Application :.

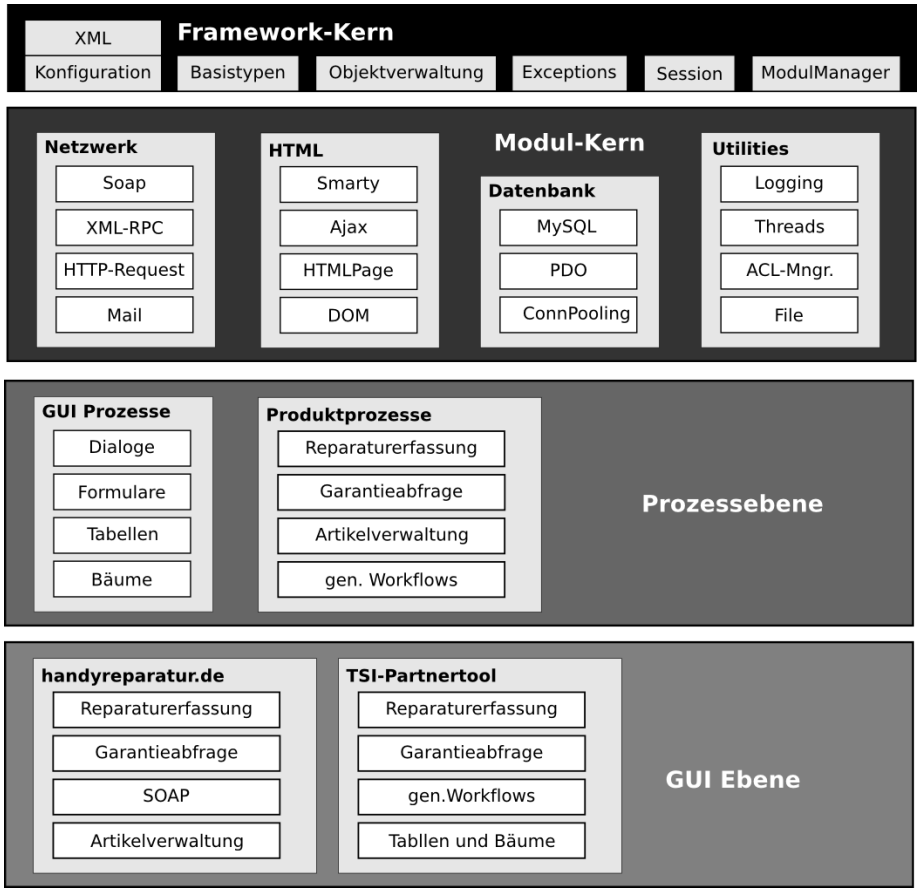
- Refactoring alter Komponenten
- Repair Management Markt
- generisches Prozess-Framework
- SOA, ALE

.: SOA und weiter? :.



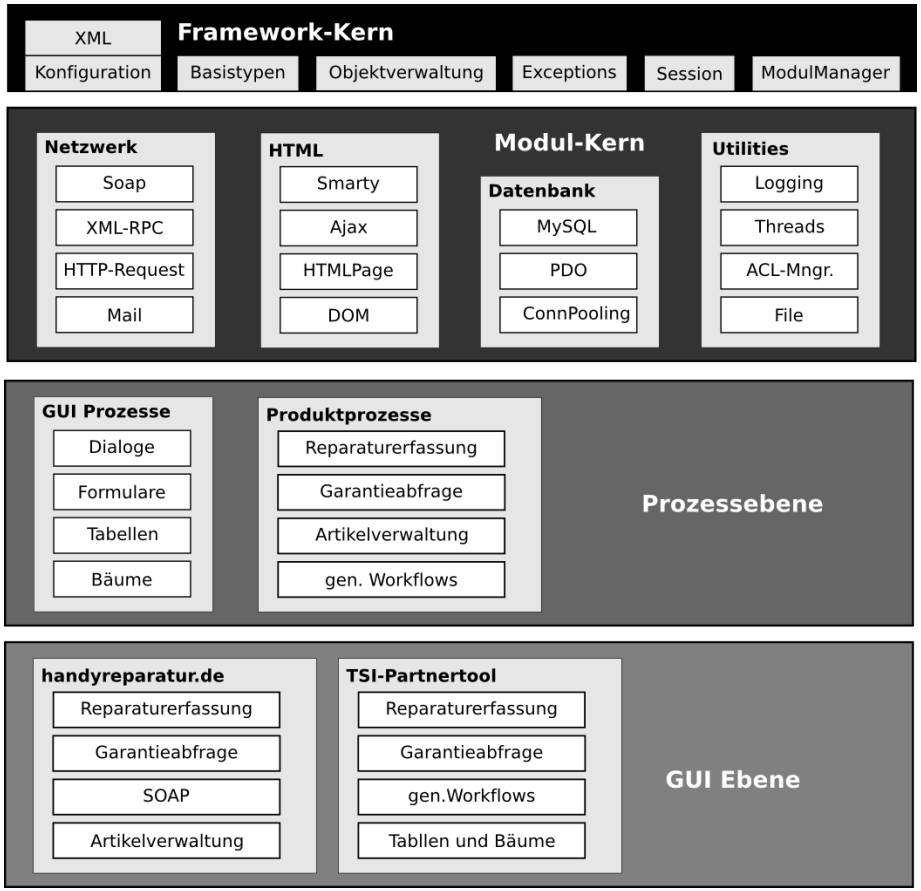
- 4 Abstraktionsebenen
- prozessorientiert

.: SOA und weiter? :.



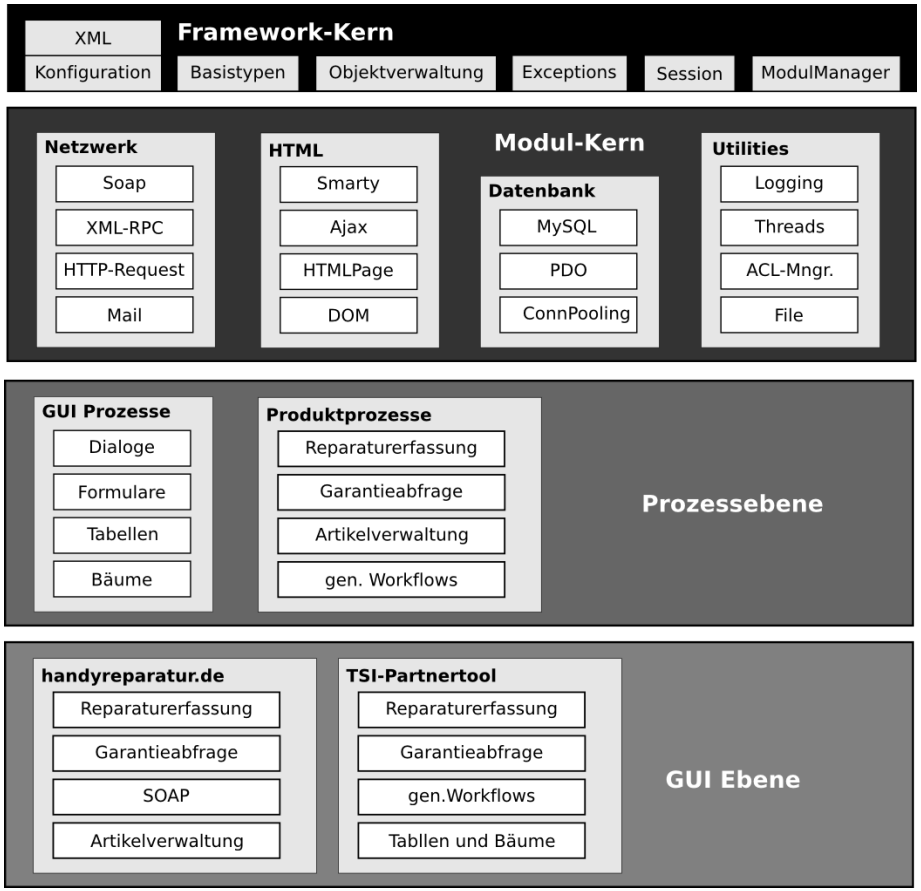
- 4 Abstraktionsebenen
- prozessorientiert
- GUI-orientiert

.: SOA und weiter? :.



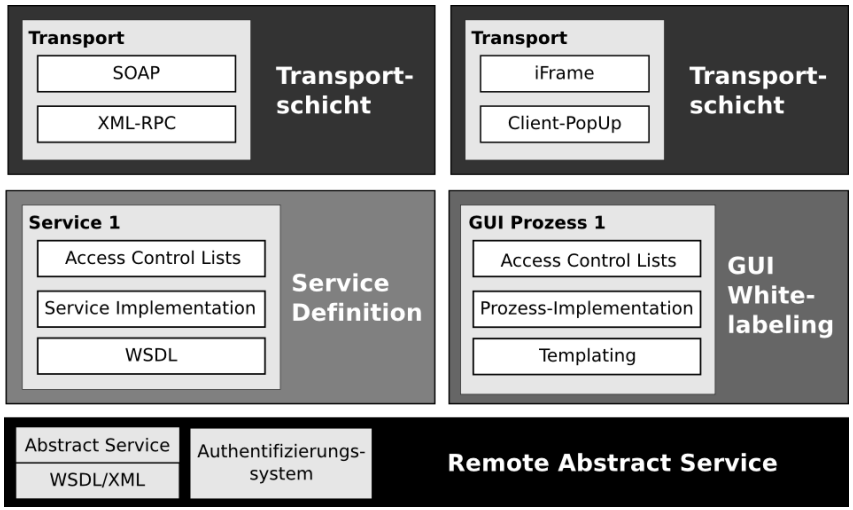
- 4 Abstraktionsebenen
- prozessorientiert
- GUI-orientiert
- alle Prozesse als Services exportierbar

.: SOA und weiter? :.



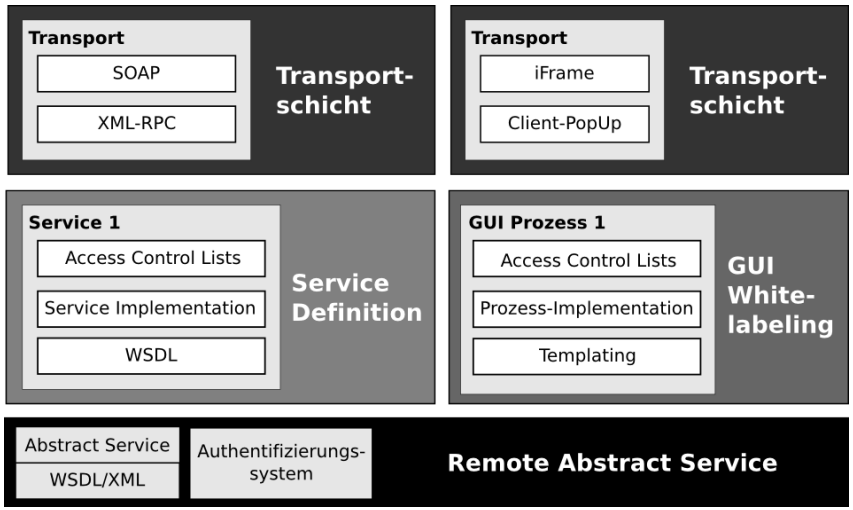
- 4 Abstraktionsebenen
- prozessorientiert
- GUI-orientiert
- alle Prozesse als Services exportierbar

.: Application Link Enabling :.



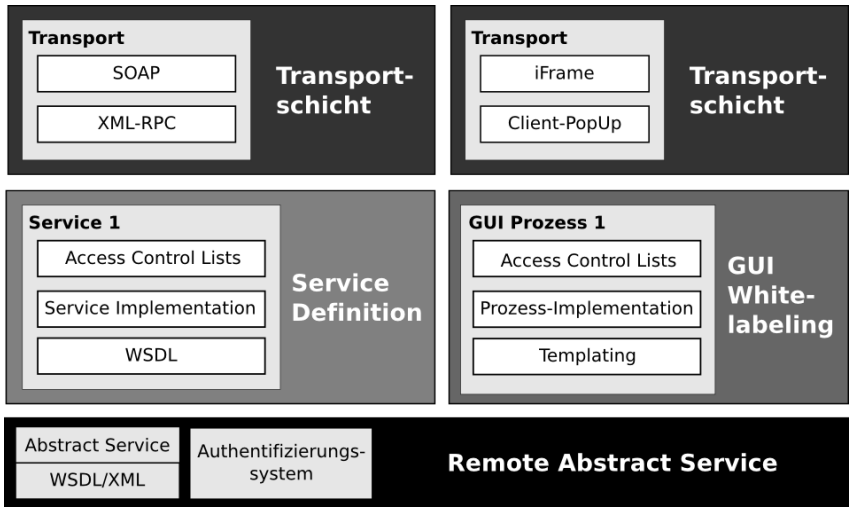
- Integration verschiedener Anwendungsdomänen
- Kompositionen exportierbar

.: Application Link Enabling :.



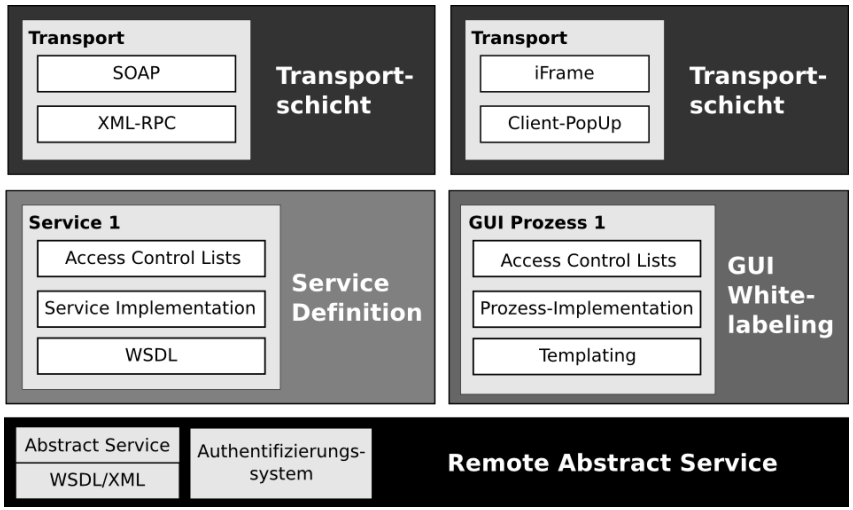
- Integration verschiedener Anwendungsdomänen
- Kompositionen exportierbar

.: Whitelabeling :.



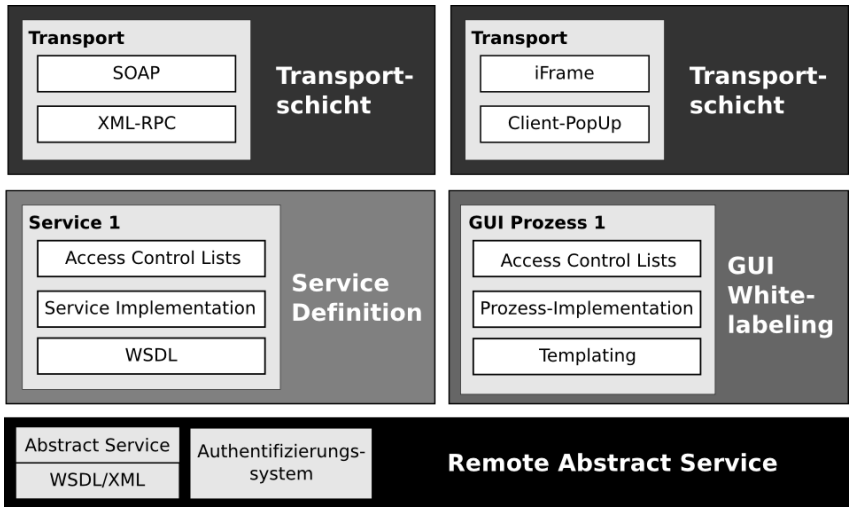
- kompositionsbezogenes. . .
  - Authentifizierungssystem
  - Templating

.: Whitelabeling :.



- kompositionsbezogenes. . .
  - Authentifizierungssystem
  - Templating
  - Prozessmanagement

.: Whitelabeling :.



- kompositionsbezogenes. . .
  - Authentifizierungssystem
  - Templating
  - Prozessmanagement

.: Fazit :.

- alle betrachteten Frameworks haben Fehler
  - starre konzeptionelle Sichtweise

.: Fazit :.

- alle betrachteten Frameworks haben Fehler
  - starre konzeptionelle Sichtweise
  - kein SOA, ALE

.: Fazit :.

- alle betrachteten Frameworks haben Fehler
  - starre konzeptionelle Sichtweise
  - kein SOA, ALE
  - oft kein Design-Pattern

.: Fazit :.

- alle betrachteten Frameworks haben Fehler
  - starre konzeptionelle Sichtweise
  - kein SOA, ALE
  - oft kein Design-Pattern
- man sollte *immer* offen für Alternativen sein

.: Fazit :.

- alle betrachteten Frameworks haben Fehler
  - starre konzeptionelle Sichtweise
  - kein SOA, ALE
  - oft kein Design-Pattern
- man sollte *immer* offen für Alternativen sein
- PHP wird unterschätzt

.: Fazit :.

- alle betrachteten Frameworks haben Fehler
  - starre konzeptionelle Sichtweise
  - kein SOA, ALE
  - oft kein Design-Pattern
- man sollte *immer* offen für Alternativen sein
- PHP wird unterschätzt

.: Diskussion :.

- Vielen Dank für Ihre Teilnahme
- Fragen und Diskussionen sind herzlich willkommen