

JÜRGEN SIEBEN, CONDES GMBH & CO. KG

**WOHIN MIT CODE IN MEINEN
APEX-ANWENDUNGEN?**

JÜRGEN SIEBEN

CONDES GMBH & CO. KG



Oracle SQL -
das umfassende Handbuch



Oracle PL/SQL -
das umfassende Handbuch

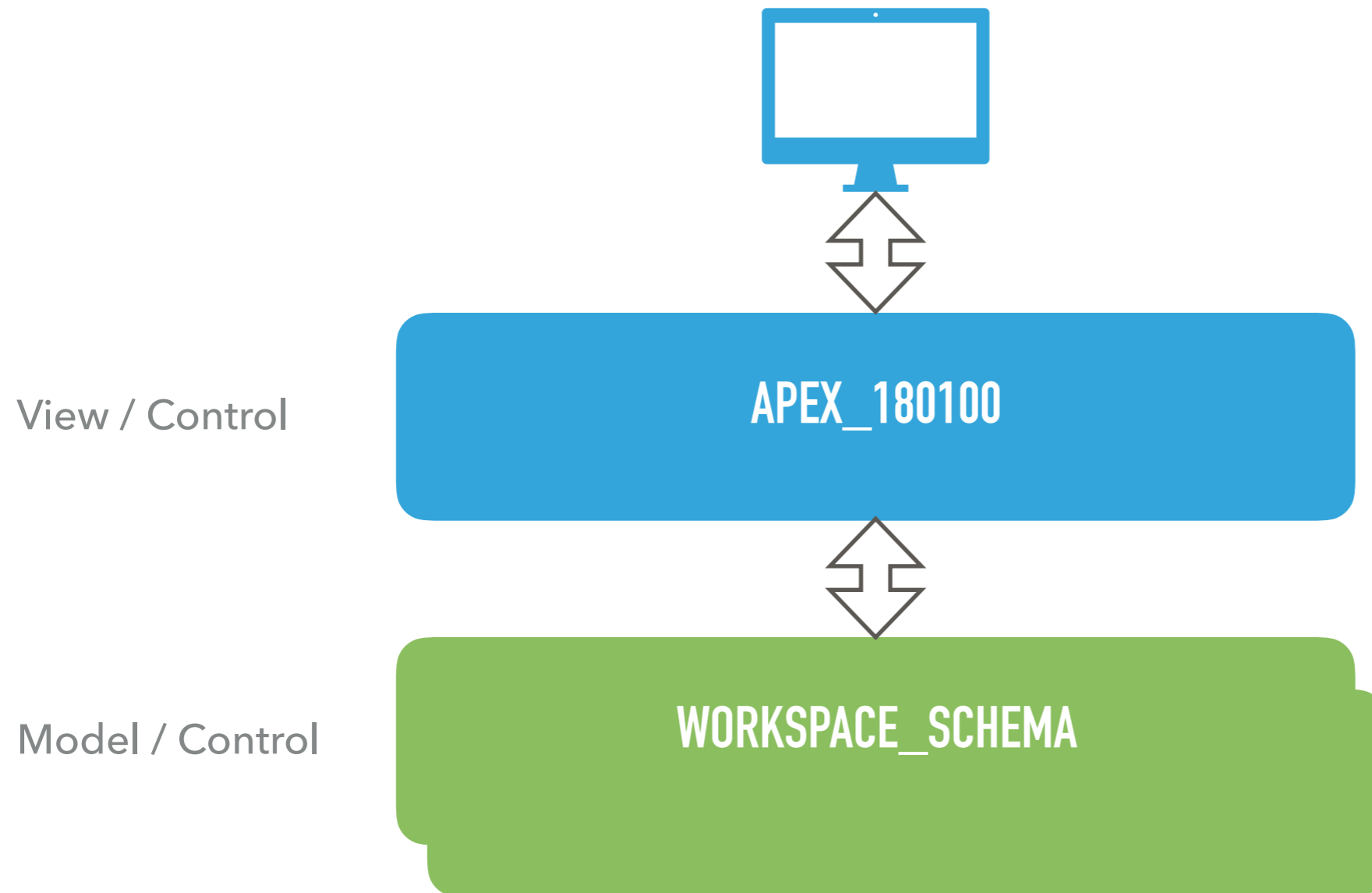


Oracle APEX -
das umfassende Handbuch

RUNDE 1:

APEX BORDMITTEL

APEX BORDMITTEL



APEX BORDMITTEL

- ▶ APEX Bordmittel sachgerecht einsetzen:
 - ▶ Berechnungen zum Vorgelegen oder Aufbereiten von Benutzereingaben
 - ▶ Validierungen zum Validieren von Benutzereingaben
 - ▶ Prozesse zur Verarbeitung von Daten
 - ▶ Verzweigungen zur Seitensteuerung
 - ▶ Verwendung der Assistenten, wo immer möglich

APEX BORDMITTEL

- ▶ Deklarative Verwendung von Validierungen und Berechnungen
 - ▶ Auslagerung längerer Code-Teile in PL/SQL-Packages
 - ▶ Auslagerung längeren JavaScript-Codes in dedizierte Datei
- ▶ Dynamische Aktionen nur für Funktionalität, die anders nicht zu erreichen ist (vs. serverseitige Bedingung)
- ▶ Vorrang von Request - Response vor AJAX-Calls

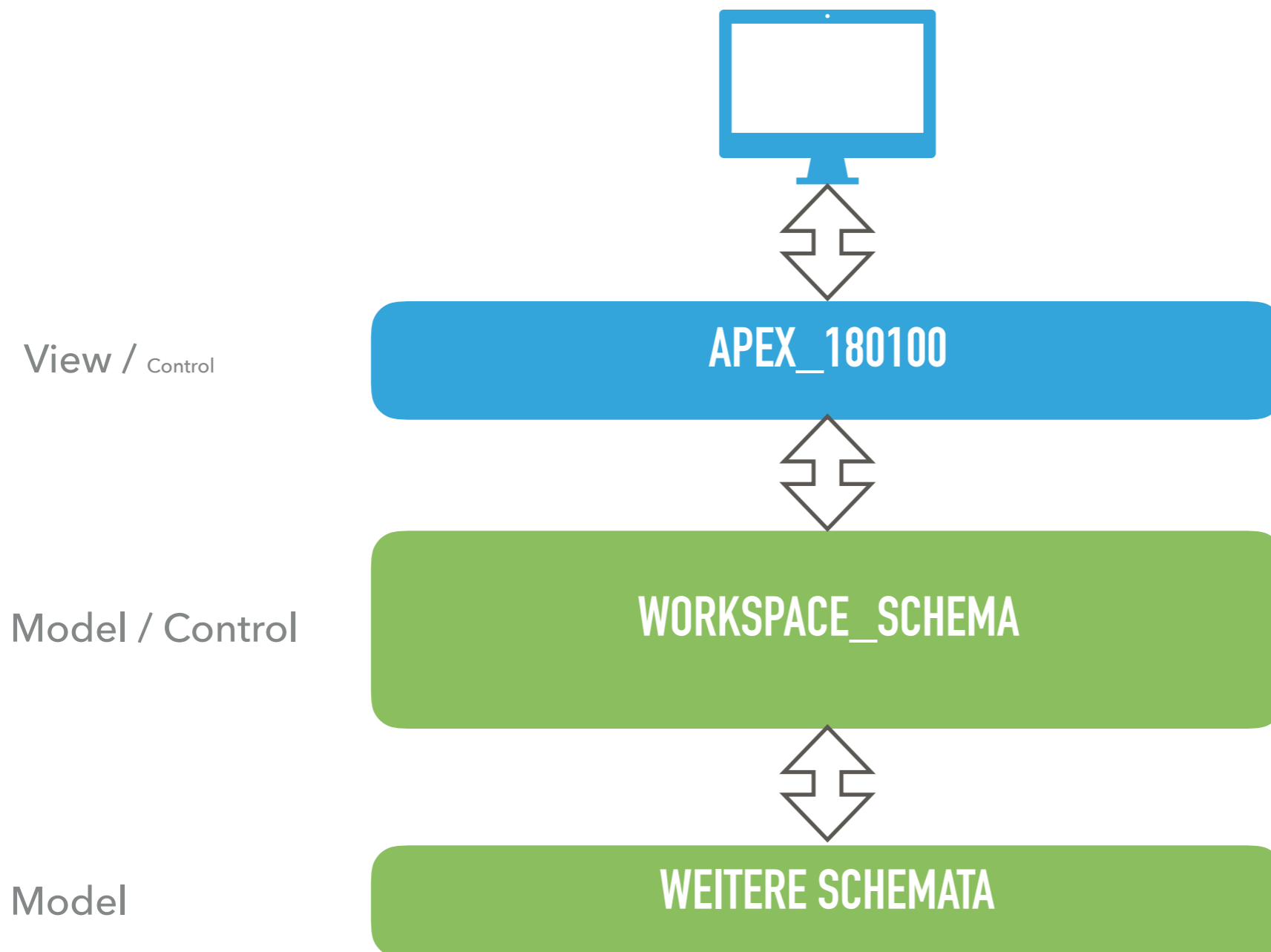
APEX BORDMITTEL – BEWERTUNG

- ▶ Vorteile
 - ▶ Deklarativ
 - ▶ »Schnell« und einfach zu entwickeln
- ▶ Nachteile
 - ▶ Vermischung von Darstellung und Logik
 - ▶ Unübersichtlich bei größeren Anwendungen
 - ▶ Schlecht migrier- und pflegbar
- ▶ Beispielanwendung: Standards-Tracker

RUNDE 2:

CLEAN APEX

CLEAN APEX



CLEAN APEX

- ▶ Kernidee: APEX-Seiten weitestgehend von PL/SQL-Code und deklarativen Validierungen entlasten
 - ▶ Bessere Kontrolle über das Geschehen
 - ▶ Einfachere Integration komplexer Validierungen
 - ▶ Einfachere APEX-Seiten
- ▶ Einführung eines dedizierten APEX-Schemas
 - ▶ Datenschemata räumen Grants auf Objekte ein
 - ▶ Nur ein Schema pro Workspace
 - ▶ Datenschemata werden von APEX-Code entlastet

CLEAN APEX

- ▶ Einführung von Kodierkonventionen
 - ▶ Lesender Zugriff über dedizierte Views pro Seite
 - ▶ Schreibender Zugriff über Package pro Seite/Seitengruppe
 - ▶ Erstellung einer Validierungs- und einer Verarbeitungsprozedur
- ▶ Offene Punkte
 - ▶ Dynamische Aktionen können zum Problem werden (Anzahl, Komplexität der Logik)
 - ▶ ...

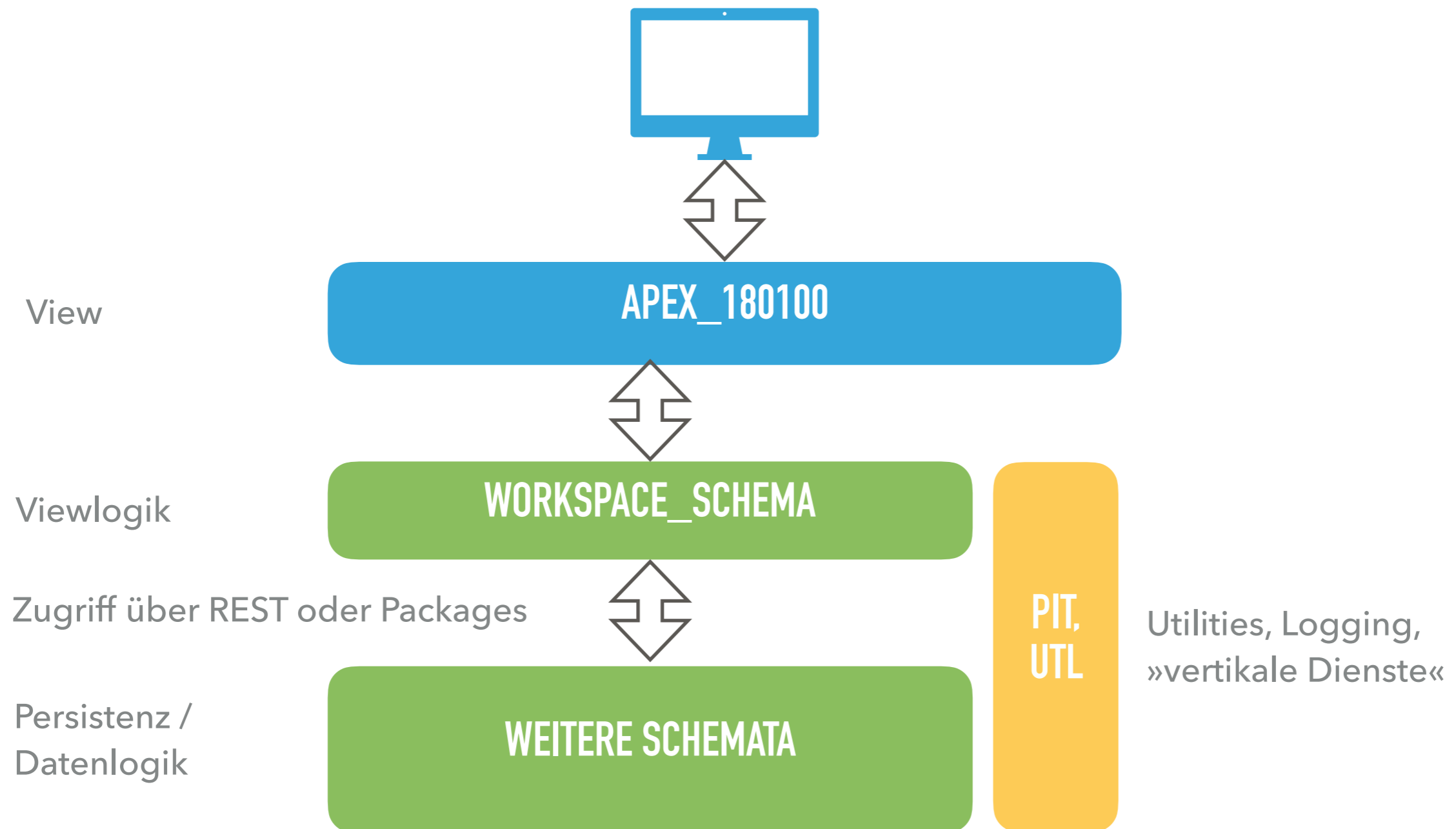
CLEAN APEX

- ▶ Offene Punkte
 - ▶ Effizienter Einsatz dieser Strategie erfordert Hilfspackages, Code-Generatoren und Code-Konventionen
 - ▶ Eindeutige Spaltenbezeichner, dedizierte Views pro Seite
 - ▶ automatisierte Übernahme der Session Status-Werte in Records
 - ▶ Trennung zwischen UI- und BL-Packages zur besseren Wiederverwendung von Code

RUNDE 3:

CLEAN CODE

CLEAN CODE



CLEAN CODE

- ▶ Kernidee: Anwendungen sind Module mit einem Interface und einer Implementierung - auch in APEX!
 - ▶ Direkter Zugriff der APEX-Oberfläche auf Tabellen nicht erlaubt
 - ▶ Trennung zwischen UI - BL - und Persistenzpackages
 - ▶ Validierung bei den Daten
- ▶ Beispiel für die Umsetzung dieses Ansatzes

CLEAN CODE

▶ Vorteile:

- ▶ Bessere Trennung der Zuständigkeiten
- ▶ Bessere Entwicklung und Wartung des Codes im Team
- ▶ Bessere Wiederverwendbarkeit für Migration auf andere Technologien

▶ Nachteile:

- ▶ Noch einmal gesteigerter Codeaufwand
- ▶ Zunächst unübersichtlicher