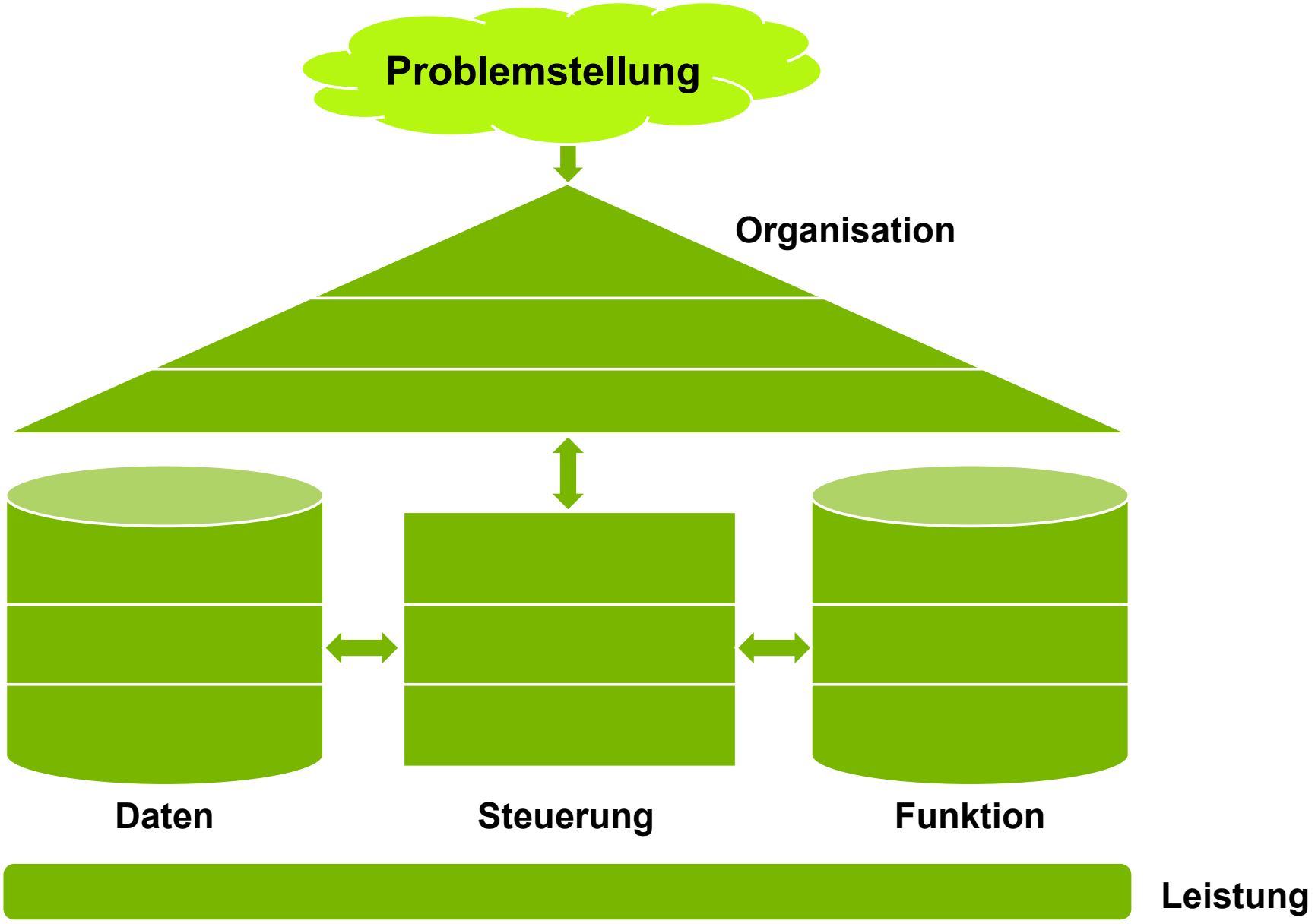
7 Datenbanken

7 Lernziele

- Sie kennen die Phasen, die ein Datenbankentwicklungs-projekt zu durchlaufen hat.
- Sie wissen, wie ein Projektteam zu bilden ist und was ein
- Lasten- und Pflichtenheft beinhaltet.
- Sie wissen, wie ein DBS in eine Applikationssoftware eingebunden wird und eine Datenbasis implementiert werden kann.

7 Einordnung ARIS-Konzept



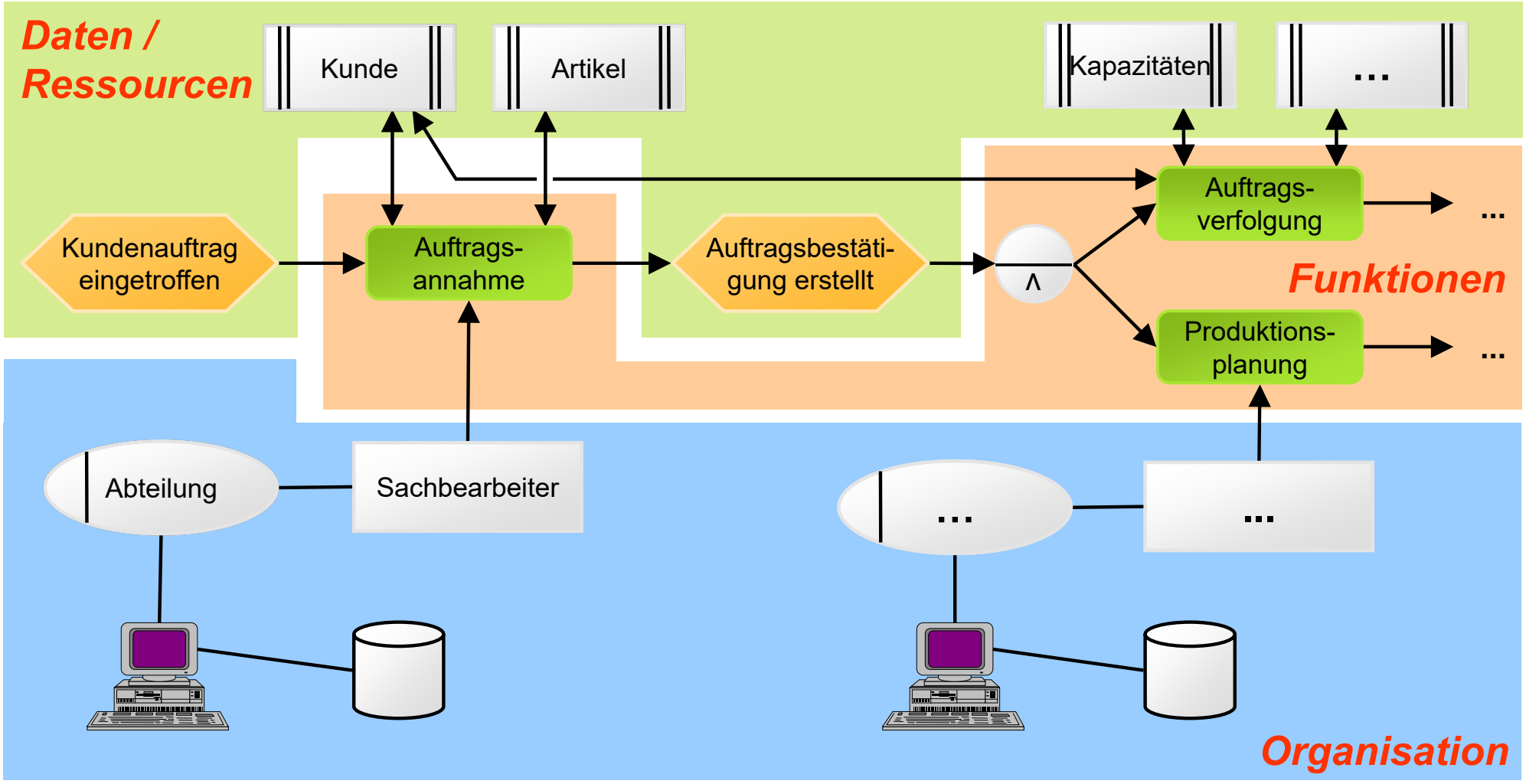
7 Einordnung ARIS-Konzept

7 Steuerungs-/Prozesssicht

Welche Prozesse im Unternehmen sollen unterstützt werden?

- Welcher Informationsfluss muss unterstützt werden?
- Welche Informationen werden benötigt?
- Welche Anforderungen gibt es von welchen Stellen?
- Wer muss auf was Zugriff haben?
- Welche Funktionen sollen unterstützt werden und wie?
- Welche Daten müssen gespeichert werden und wie hängen diese zusammen?
- Welche Ressourcen sind verfügbar und müssen/können mit dem System gekoppelt werden?

Ereignisgesteuerte Prozesskette (EPK)



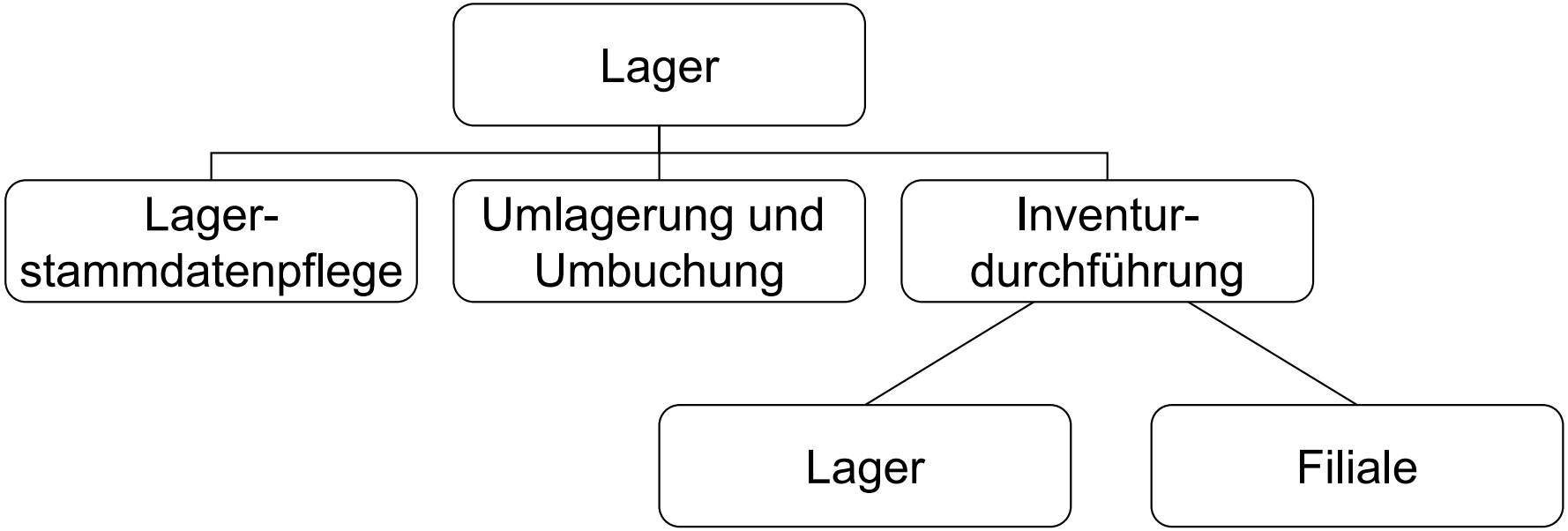
7 Einordnung ARIS-Konzept

7 Funktionssicht

Welche Ereignisse/Vorgänge lösen welche „Funktionen“ aus?

- Welche Funktionen werden benötigt?
- Stellt die hierarchische Beziehungen der verschiedenen Funktionen zueinander dar (Funktion → Teilfunktionen)
- Funktionen erzeugen, verwenden oder manipulieren Informationsobjekte, die in Ereignissen definierte Zustände erhalten.

Funktionsbaum



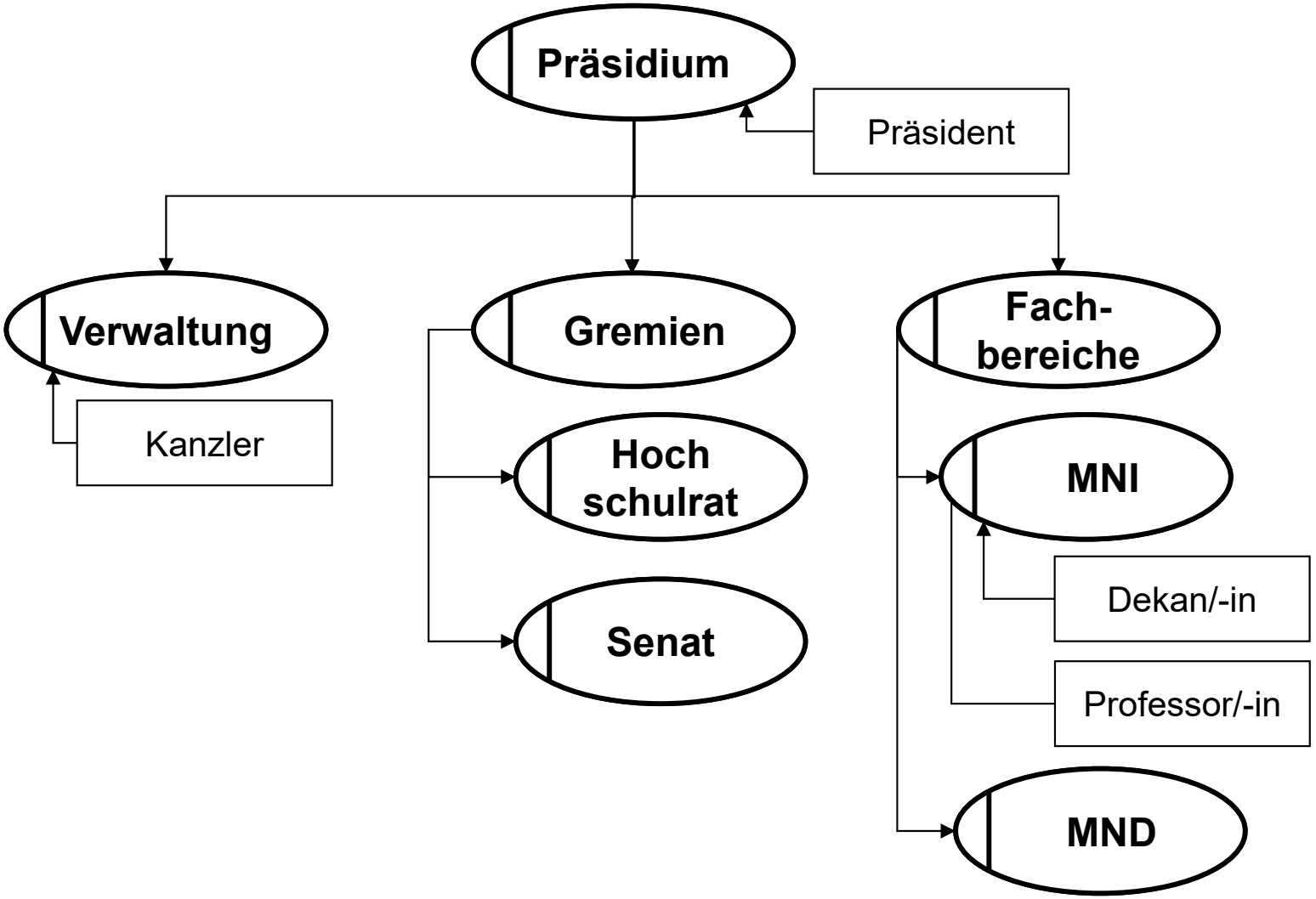
3 Einordnung ARIS-Konzept

7 Organisationssicht

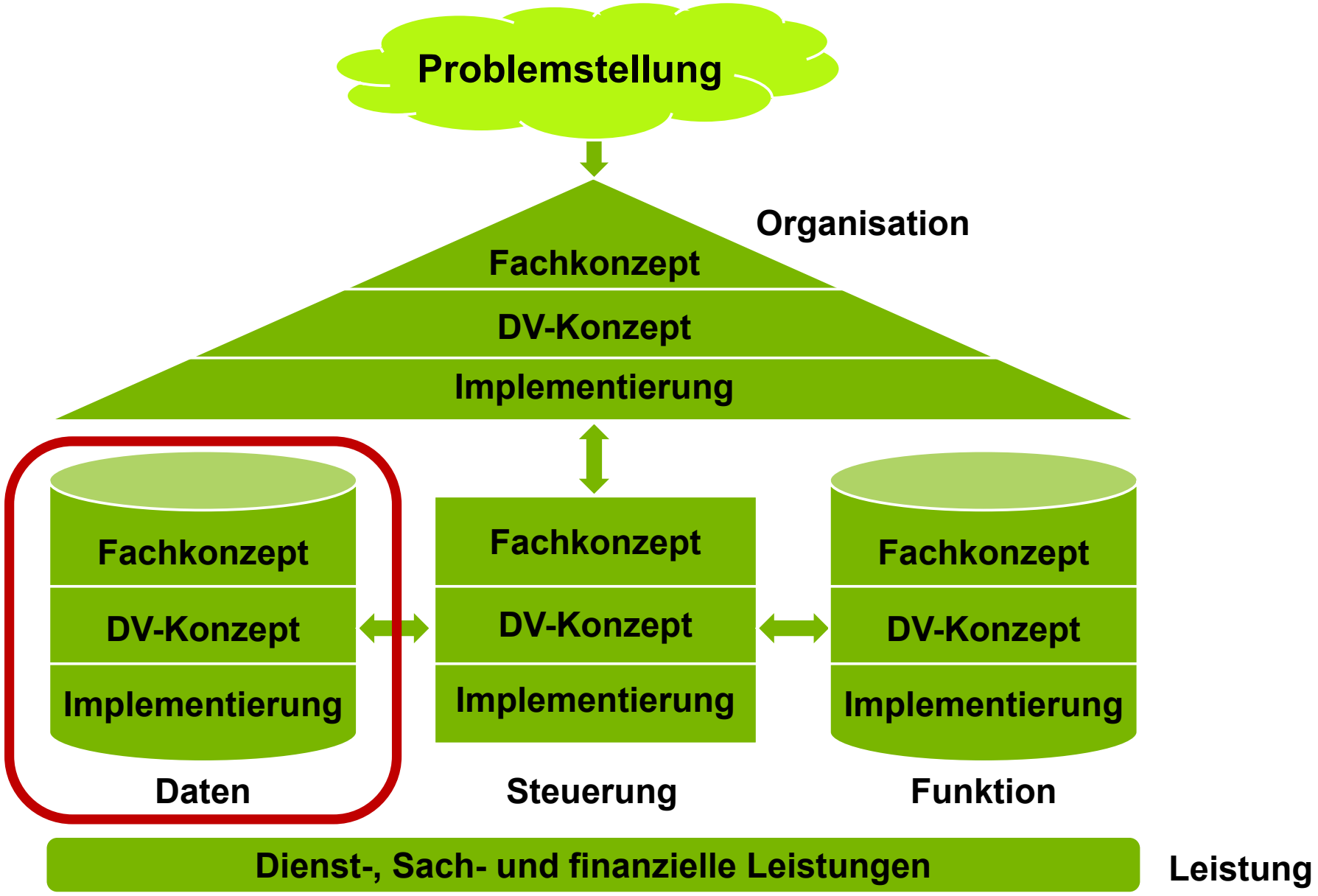
Folgende Fragen soll ein Organigramm beantworten:

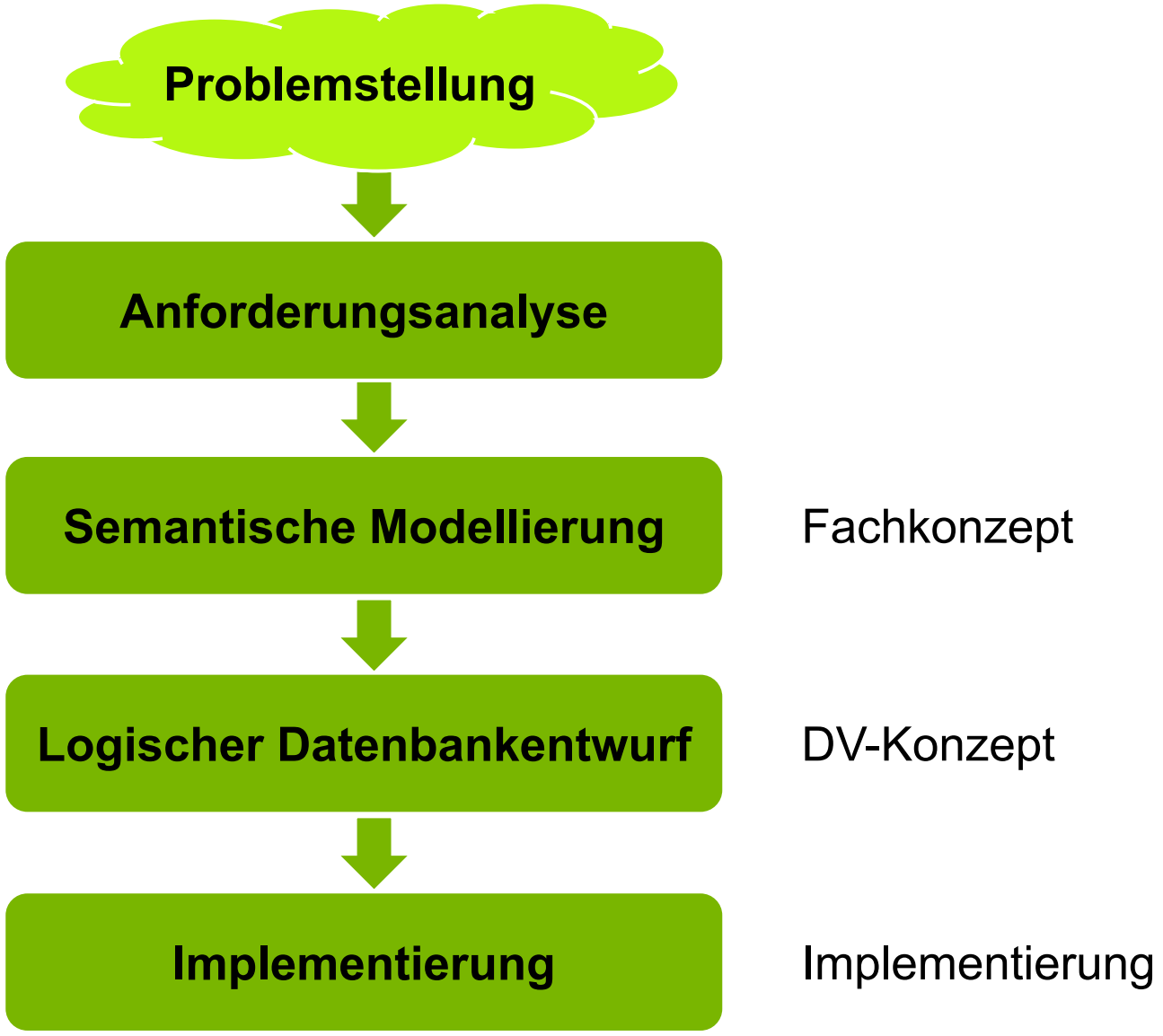
- Welche Teile des Unternehmens stehen in Verbindung mit welchen anderen?
- Wie sollen Menschen und Prozesse zusammenwirken?
- Welche Aufgaben und Abläufe gibt es?
- Wer darf wem Aufgaben delegieren und Anweisungen erteilen?
- Wer trägt welche Verantwortung und ist haftbar, wenn es um rechtliche Vorgaben geht?
- Wer darf auf welche technischen Systeme zugreifen (mit entsprechenden Benutzungsrechten und Zugangsdaten)?

Organigramm



3 Einordnung ARIS-Konzept




Problemstellung
Anforderungsanalyse
Semantische Modellierung
Fachkonzept
Logischer Datenbankentwurf
DV-Konzept
Implementierung
Implementierung

7 Anforderungsanalyse

Statische Anforderungen (Datenstruktur)

- Entitytypen (Kunden, Lieferanten usw.)
- Beziehungstypen (Kunde hat Auftrag)
- Attribute (Kunden(Kunr., Kuname usw.)
- Attributseigenschaften (numeric, alpa usw.)

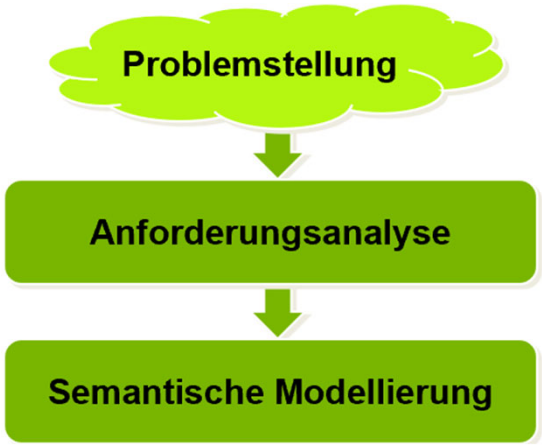
Dynamische Anforderungen

- Festlegung der auszuführenden Operationen
- Benutzerhäufigkeit und Häufigkeit des Datenanfalls
- Zugriffs- bzw. Zugangsbestimmungen
- Anforderungen an die Geschwindigkeit
- Sicherheits- und Schutzanforderungen

Systemanalysemethoden (verbale und bildliche Beschreibung)

- Unterlagenstudium, Fragebogen, Interviews, Selbstaufschreibung, Beobachtung

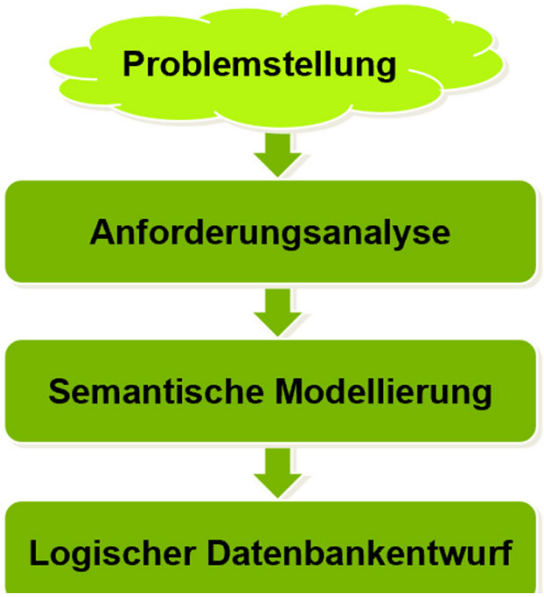
7 Semantische Modellierung



- Datenbankunabhängiger Entwurf durch modellhafte Beschreibung der analysierten Daten
- Methode: ERM, SERM, UML

- Zunächst globales, grobes Datenmodell, das die Informationsanforderungen aller späterer Datenbanknutzer berücksichtigt
- Dann schrittweise Verfeinerung und Anpassung an spezielle Anforderungen
- Ergebnis: Konzeptionelles Datenmodell

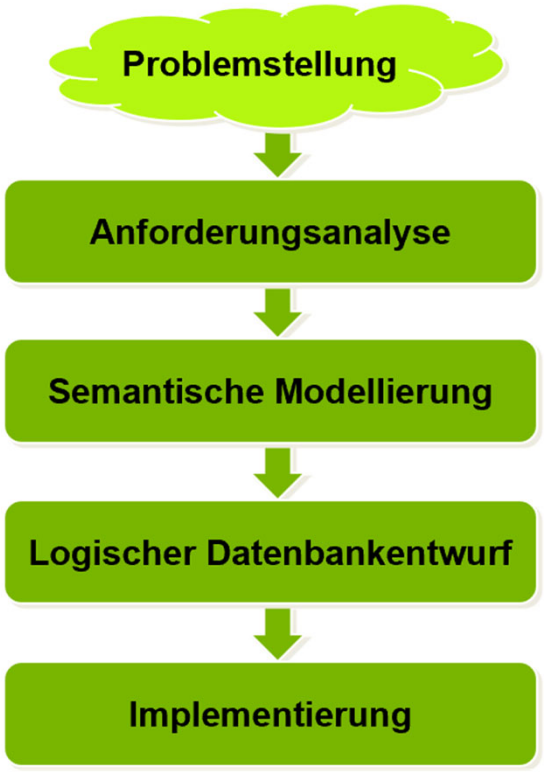
7 Logischer Datenbankentwurf

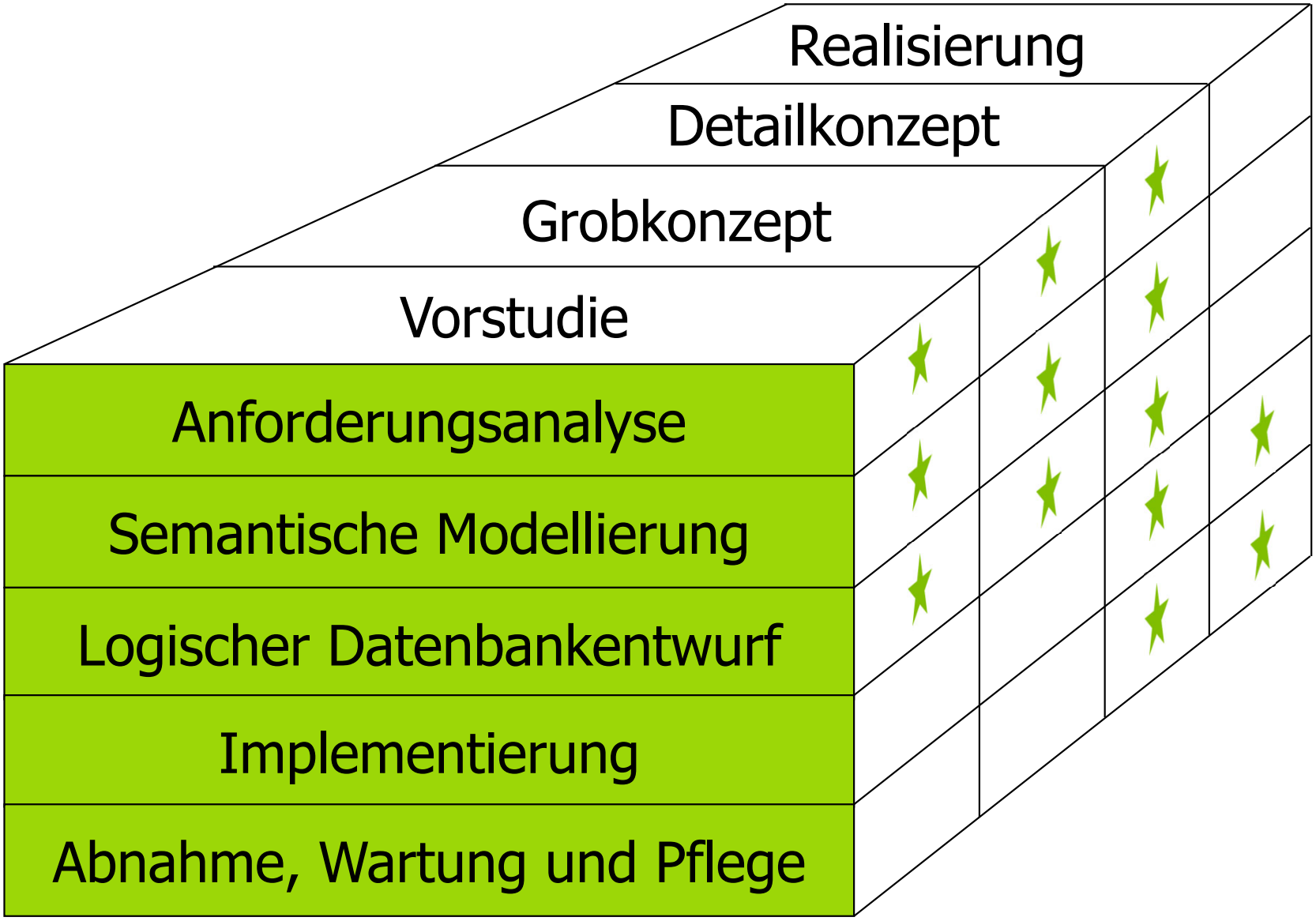


- Datenbankabhängiger, modellhafter Entwurf
- Entwurfstypen: Hierarchisches Modell, Netzwerkmodell, Relationenmodell, Objektmodell, etc.
- Methode: z.B. Normalisierung bei Relationenmodellen

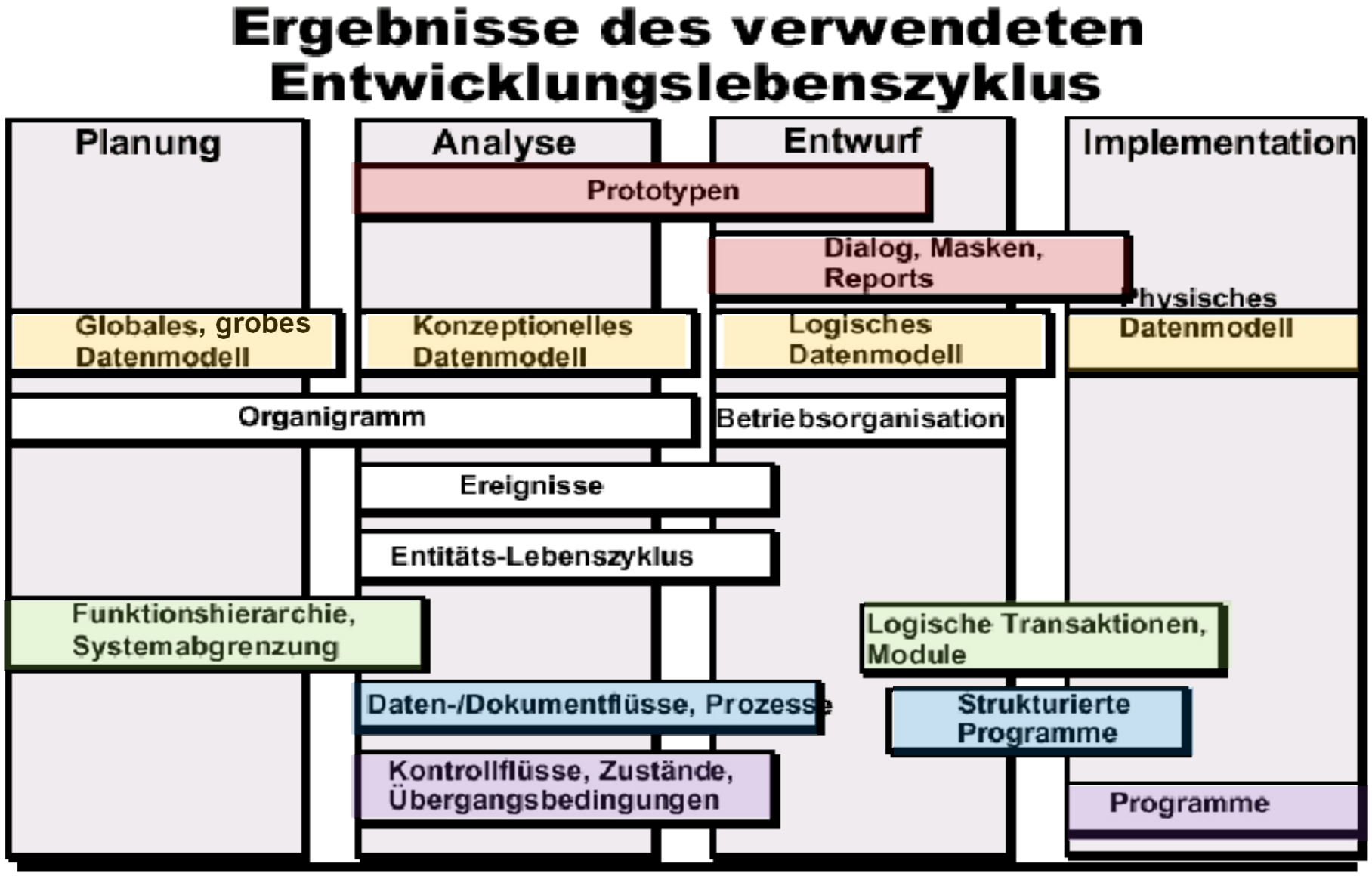
7 Implementierung

- Umsetzung des konzeptionellen Schemas
- Berücksichtigung der Benutzerzugriffs-rechte
- Festlegen der Speicherparameter, der physischen Datenstruktur und der Datenbankparameter
- Übernahme von Daten



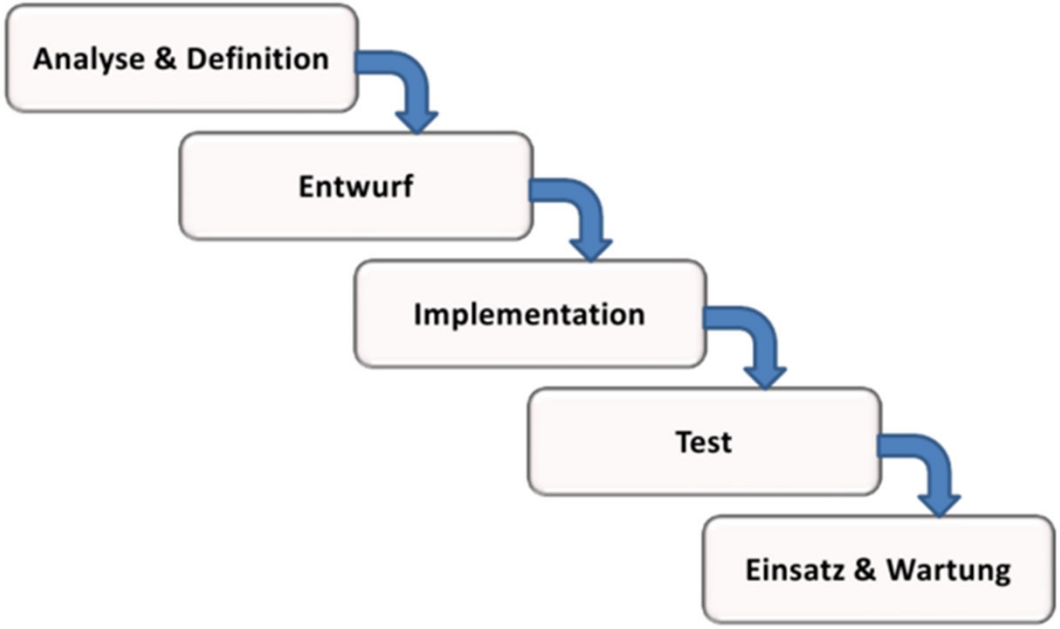

Realisierung
Detailkonzept
Grobkonzept
Vorstudie
Anforderungsanalyse
Semantische Modellierung
Logischer Datenbankentwurf
Implementierung
Abnahme, Wartung und Pflege

Systementwicklung im Überblick



7 Klassisches Vorgehensmodell

- 5 sequentielle Phasen
- Eine Phase muss vollständig abgeschlossen sein bevor die nächste beginnt.
- Jeder Phase liefert Entwicklungsergebnisse in Form von Anforderungs- bzw. Entwurfsdokumenten (Meilensteine).
- Anzahl und Inhalt der Phasen sind je nach Modell unterschiedlich.



Lasten- und Pflichtenheft

Lastenheft
Spezifikation der Anforderungen im Hinblick auf das zu realisierende System durch den Auftraggeber (Benutzer/Fachspezialisten)

Pflichtenheft
Detaillierte und konsistente Dokumentation der Art und Weise, wie der Auftragnehmer die Anforderungen aus dem Lastenheft konkret erfüllt

7 Lastenheft

- Anforderungen an das Projekt unabhängig von Realisierung
- Alle Basisanforderungen und grobes Konzept des Projekts, keine exakten Anforderungen
- Zusammenfassung wirtschaftlicher, technischer und organisatorischer Erwartungen
- Erwarteter Umfang des Projektes

- Einfachste Form: Liefertermin und Preis

7 Pflichtenheft

- Basiert auf dem Lastenheft
- Definition der Realisierung der Anforderungen
- Beschreibung, wie und womit das Produkt entwickelt wird (Mock-ups, gewünschte Reports, etc.)
- Detaillierte Beschreibung der technischen und organisatorischen Anforderungen
- Grundlage zur Auswahl der Angebote für den Auftraggeber

7 Zweck von Lasten- und Pflichtenheft

- Für beide Seiten verständliches Profil des Projekts
- Vorbeugung von Missverständnissen
- Zeit- und Kostenersparnis, da Inhalt des Projekts **vor** Entwicklung festgehalten wird und so weniger Nachbesserungen notwendig sind

7 Projektbeteiligte

- Auftraggeber
  - „Bestellt“ und bezahlt die Leistungen aus dem Projekt
- Lenkungsausschuss (Entscheidungsgremium)
  - Oberste Instanz des Projektes, bestehend aus Entscheidungsträgern aller beteiligten Bereiche (intern und extern)
  - Weichenstellung im Projekt
- Projektleiter (PL)
  - Aufgabenverteilung
  - Koordination des Projektteams und mit dem Lenkungsausschuss
  - Ressourcenüberwachung
  - Projektcontrolling
- Projektteam
  - Durchführung und Verantwortung für einzelne Aufgaben

7 Projektbeteiligte

- Prozess- bzw. funktionsverantwortliche Personen
  - Testen und prüfen nach erfolgreicher Umsetzung
- Betroffene
  - Mitwirkung in Arbeitsgruppen, Workshops, Befragungen
- Funktional Beteiligte
  - Beratung im Projekt, Interessenvertretung, Wahrnehmung gesetzlicher Aufgaben
- Sponsor
  - Steht mit seiner Autorität hinter dem Projekt

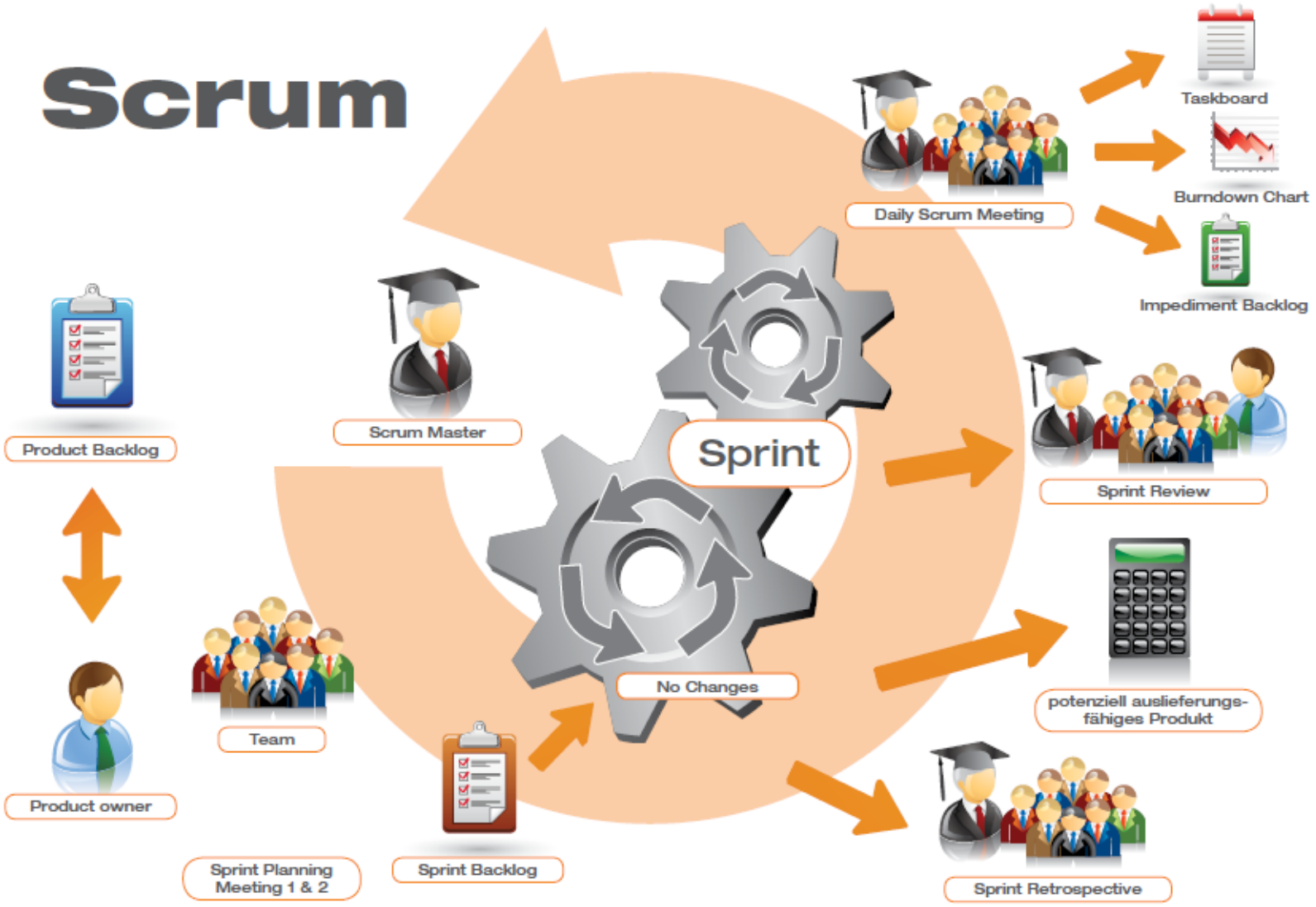
7 Projektbeteiligte

Bei jedem Projekt gibt es Personen, die ihre Haut zu Markte tragen, also tatkräftig mitarbeiten und auch ein echtes Risiko auf sich nehmen. Sie sind „committed". In Scrum werden solche Personen daher „Pigs“ genannt.

Auf der anderen Seite findet man auch immer solche Personen, die zwar gern mitreden, kritisieren und schlussendlich am Erfolg partizipieren wollen, ansonsten aber eher unbeteiligt den eigentlichen Machern bei der Arbeit zusehen. Sie wollen gern „involved“ sein. Diese Personen werden in Scrum daher „Chickens“ genannt.

A chicken and a pig were brainstorming...

- **Chicken:** Let's start a restaurant!
- **Pig:** What would we call it?
- **Chicken:** Ham 'n' Eggs!
- **Pig:** No thanks. I'd be committed, but you'd only be involved!


Scrum
Taskboard
Daily Scrum Meeting
Burndown Chart
Impediment Backlog
Product Backlog
Scrum Master
Sprint
Sprint Review
No Changes
potenziell auslieferungs-
fähiges Produkt
Team
Product owner
Sprint Planning
Meeting 1 & 2
Sprint Backlog
Sprint Retrospective

Projektmanagement

Projekt
Erledigung einer einmaligen, neuartigen Aufgabe mit definiertem Ziel von i.d.R. hoher Komplexität bei begrenzten Ressourcen und hoher Interdisziplinarität

Management
Planung, Steuerung und Kontrolle

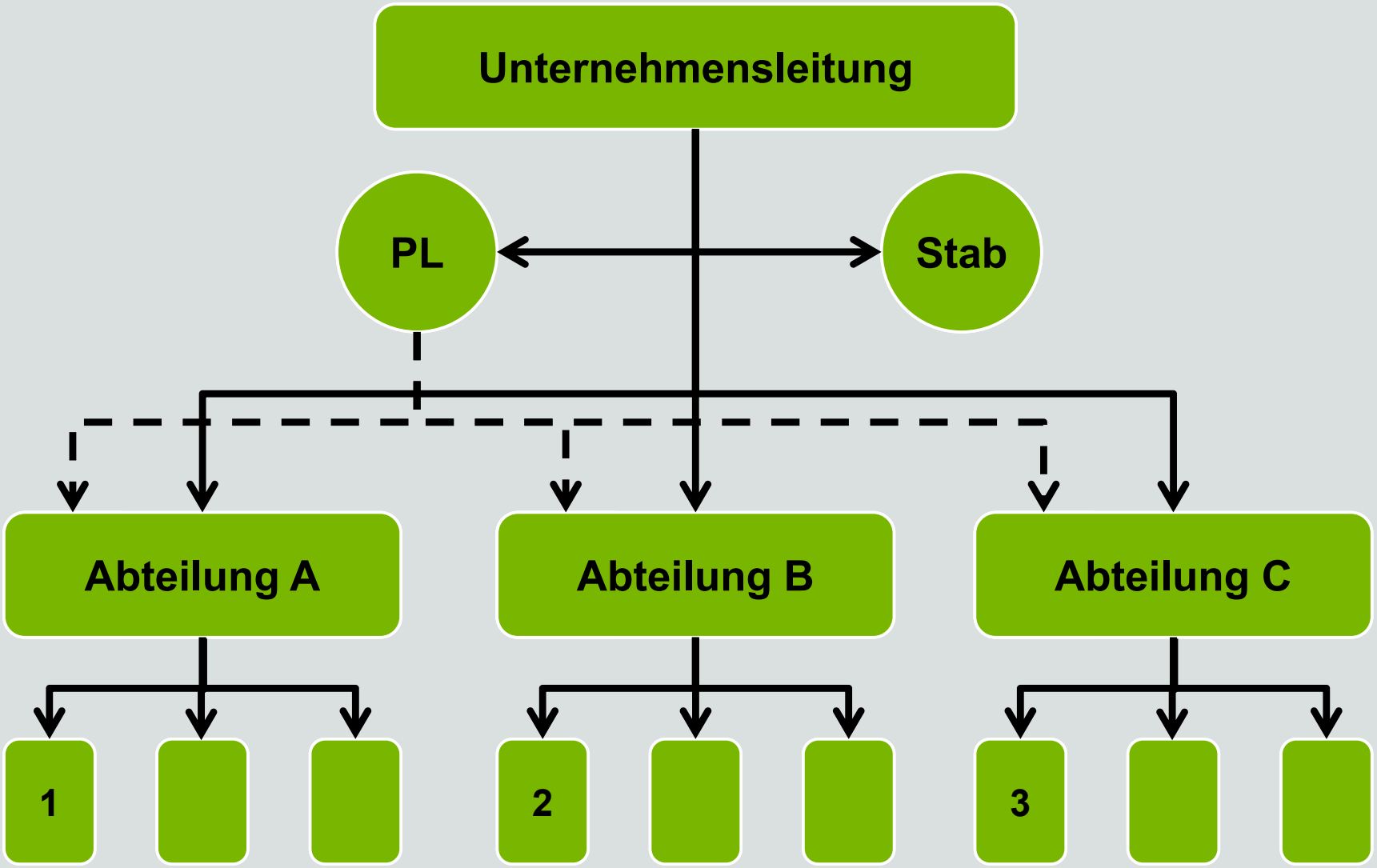
Projektmanagement
Planung, Steuerung und Kontrolle zur zielgerichteten Abwicklung eines Projektes

7 Projektteam

7 Projektorganisation

- Stabs-Projektorganisation
  (Einfluss-Projektorganisation)
- Reine Projektorganisation
  (unabhängige Projektorganisation oder Task Force)
- Matrix-Projektorganisation

7 Stabs-Projektorganisation



7 Stabs-Projektorganisation

Charakteristika

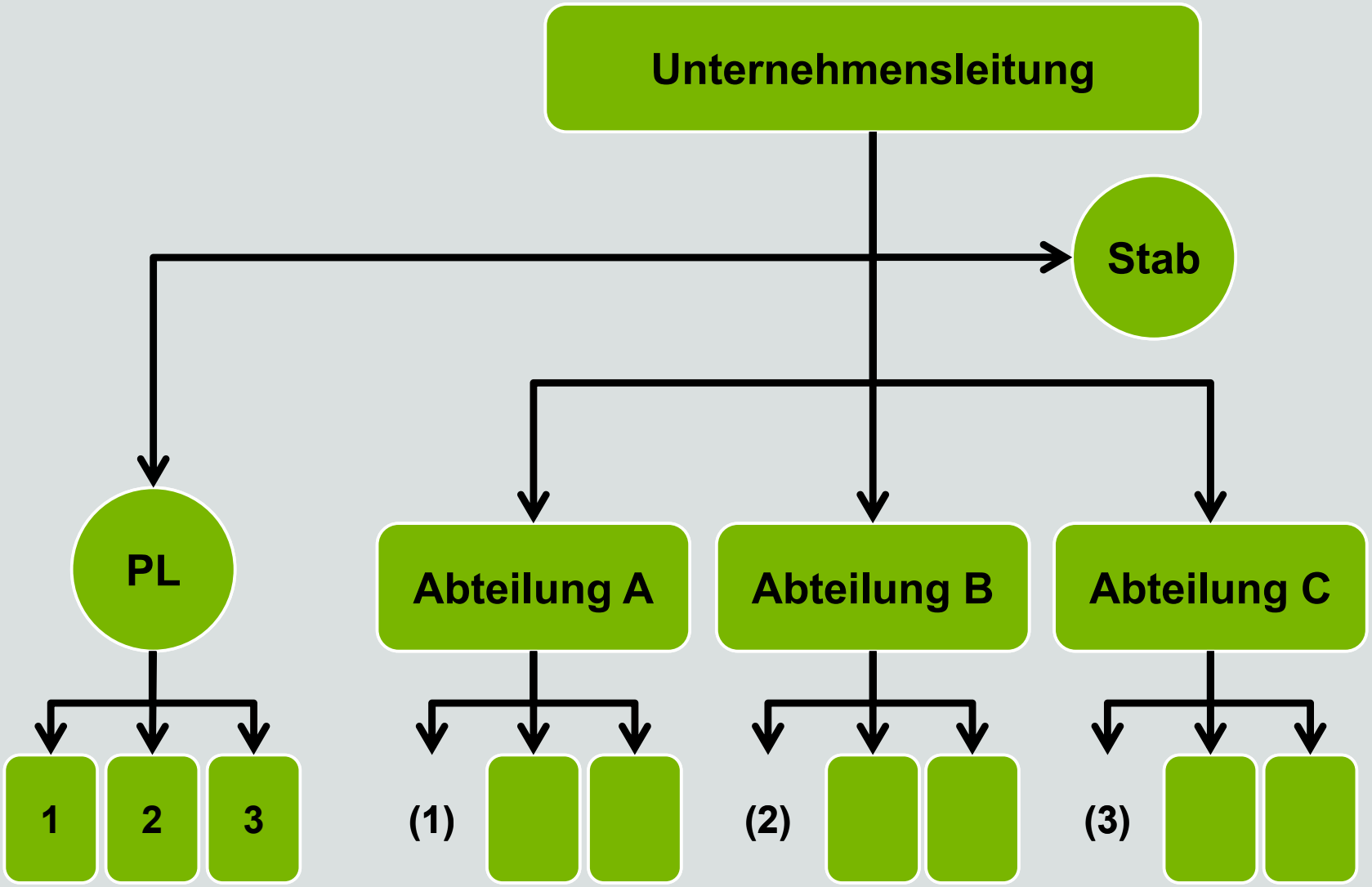
- Funktionale Hierarchie im Unternehmen unverändert
- Projektleiter (PL)…
  - übt nur eine beratende, koordinierende und entscheidungsvorbereitende Funktion aus
  - hat keine Entscheidungs- und Weisungsbefugnisse
  - plant, überwacht und steuert den Projektablauf in sachlich, terminlicher und kostenmäßiger Hinsicht
  - schlägt Maßnahmen vor
  - kann für die Nichterreichung der Projektziele nicht (allein) verantwortlich gemacht werden
  - ist verantwortlich für die rechtzeitige Information der Leitungsinstanzen sowie für die Qualität seiner Vorschläge
  - ist im Allgemeinen sehr hoch in der Hierarchie angesiedelt

Stabs-Projektorganisation

Anwendungsbedingungen

- Projekt betrifft mehrere Geschäftseinheiten, jedoch nur punktuell, so dass ein Freistellen von Mitarbeitern nicht gerechtfertigt wäre
- Leistungen der Projektmitarbeiter können getrennt erbracht werden. Koordination erfolgt zu Jour-Fixe-Terminen
- PL besitzt hohe persönliche und fachliche Autorität und hat eine starke Position im Mitarbeiterbewusstsein
- Projekt ist in einer frühen Phase
- Projektrelevanz ist für das Unternehmen gering

7 Reine Projektorganisation



7 Reine Projektorganisation

Charakteristika

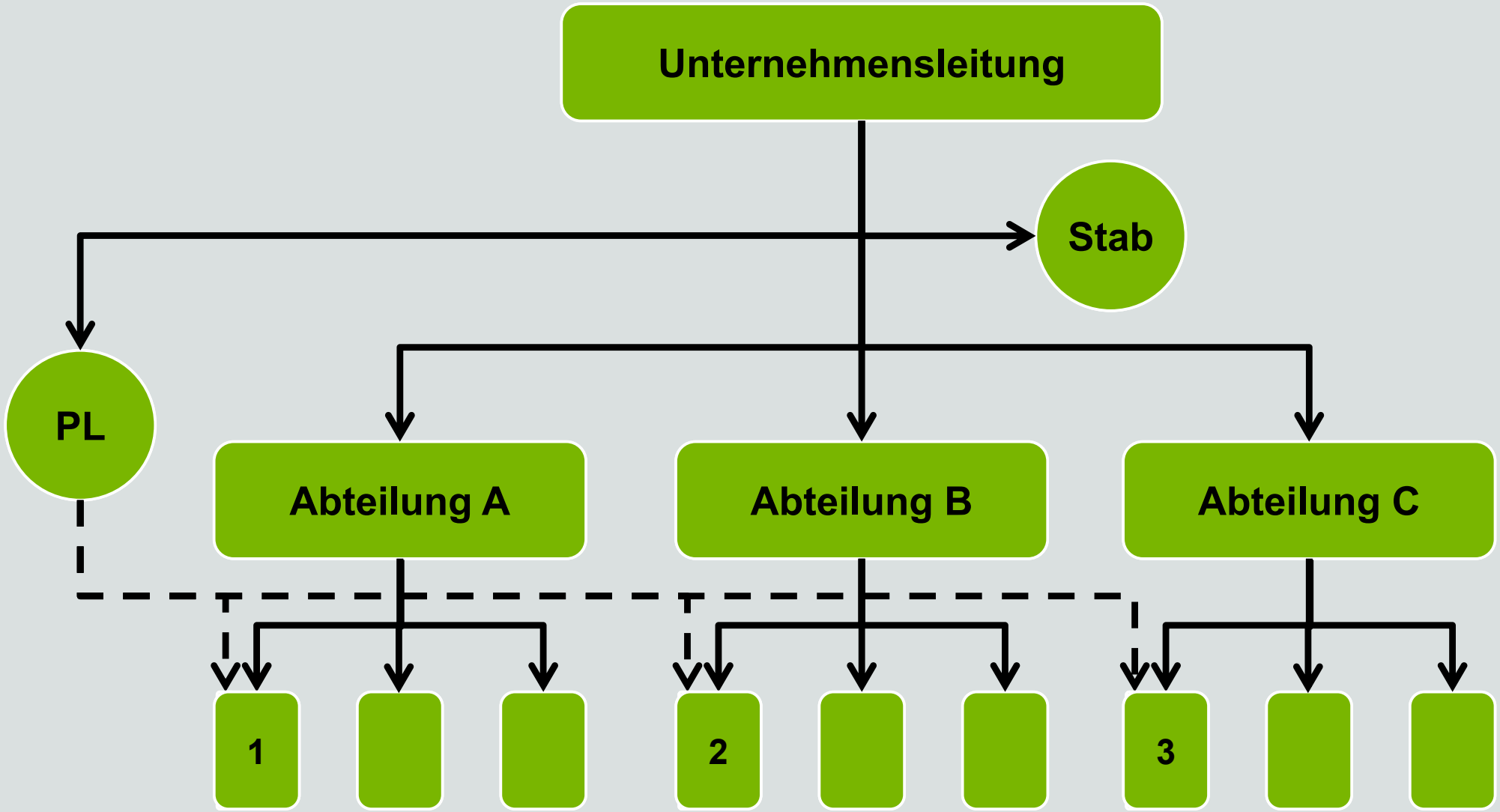
- Projektleiter
  - hat volle Kompetenzen in Bezug auf Projektmitarbeiter, Betriebsmittel und Budget
  - lediglich grundlegende disziplinarische Befugnisse Vergütung, Fortbildung, Bewertung usw. bleiben beim Linienvorgesetzten
  - besitzt volle Projektverantwortung
- In der Regel liegt eine Vollzeit-Freistellung der Mitarbeiter für das Projekt vor.

Reine Projektorganisation

Anwendungsbedingungen

- Projektziele und Umfang sind klar definiert
- Umfangreiches, sehr wichtiges und dringendes Projekt, das die Freistellung von Mitarbeitern rechtfertigt
- Stellvertreterprobleme aufgrund der Freistellung sind lösbar
- Klare Personalplanung, so dass erkennbar ist, was die freigestellten Mitarbeiter nach dem Projekt tun
- PL ist ausreichend qualifiziert und angesehen, so dass auch hierarchisch gleichgestellte Mitarbeiter zugeordnet werden können

Matrix-Projektorganisation



7 Matrix-Projektorganisation

Charakteristika

- Kombination von Stabs- und reiner Projektorganisation, d.h. eine beliebige Organisation einer Unternehmung wird durch zusätzliche projektbezogene Weisungsrechte überlagert.
- Der PL ist für die Projektplanung, -entscheidung, -steuerung und -kontrolle (was?, wann?) verantwortlich; für die projektbezogenen fachlichen Aufgaben (wie?) sind die Linieninstanzen verantwortlich.
- Projektmitarbeiter werden temporär in das Projektteam delegiert; unterstehen fachlich dem PL, disziplinarisch dem Linienvorgesetzten

Matrix-Projektorganisation

Anwendungsbedingungen

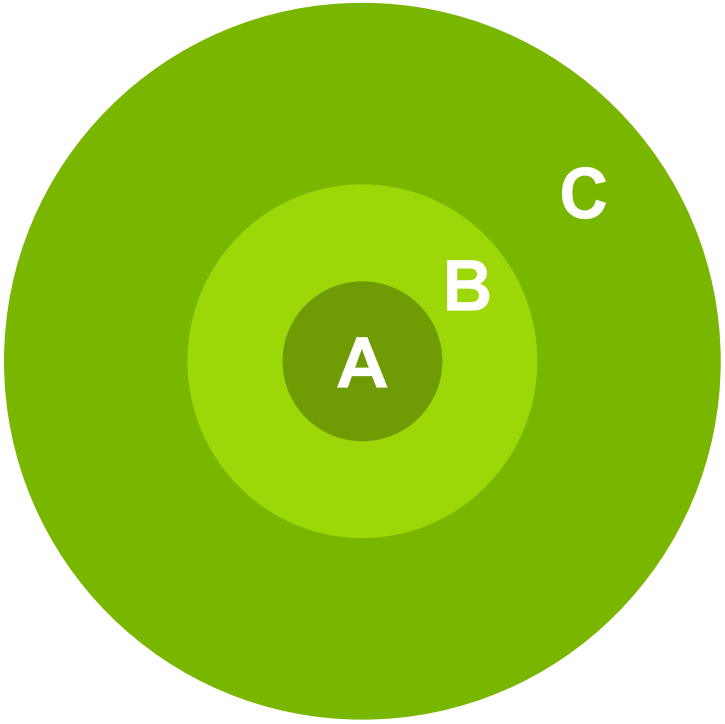
- Projekt ist in relativ klare Arbeitspakete gliederbar, die auch getrennt bearbeitet werden können
- Kleine bis mittlere Komplexität des Projekts
- PL hat relativ starke formale und/oder informale Stellung
- In den Fachabteilungen sind die notwendigen Kapazitäten vorhanden
- Es existiert ein gut entwickeltes Organisations- und Führungsverständnis aller Beteiligten

7 Entwicklungsaufwand eines Datenbankprojektes

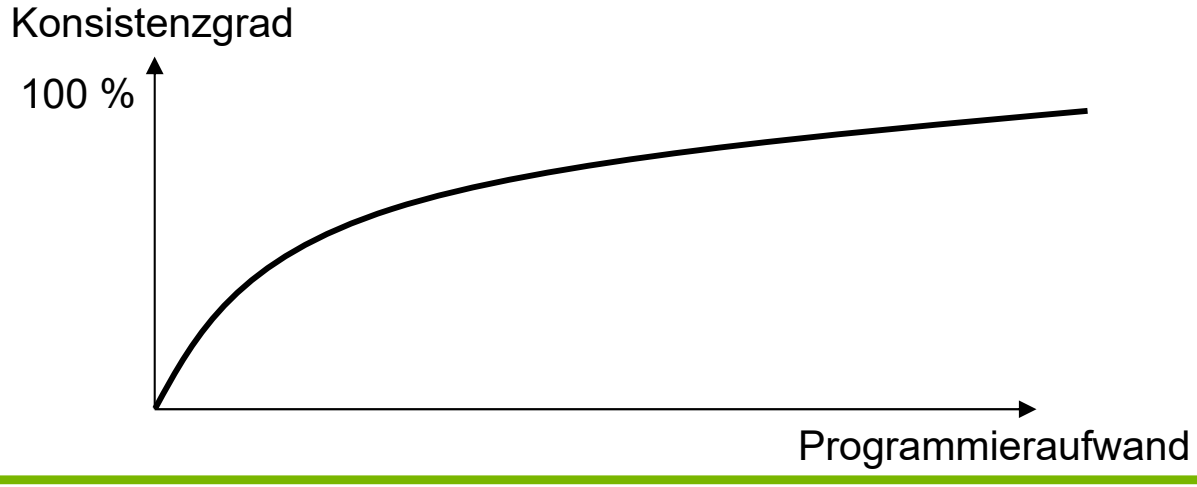
A = Datenbasis einrichten

B = Zugriffsberechtigungen vergeben

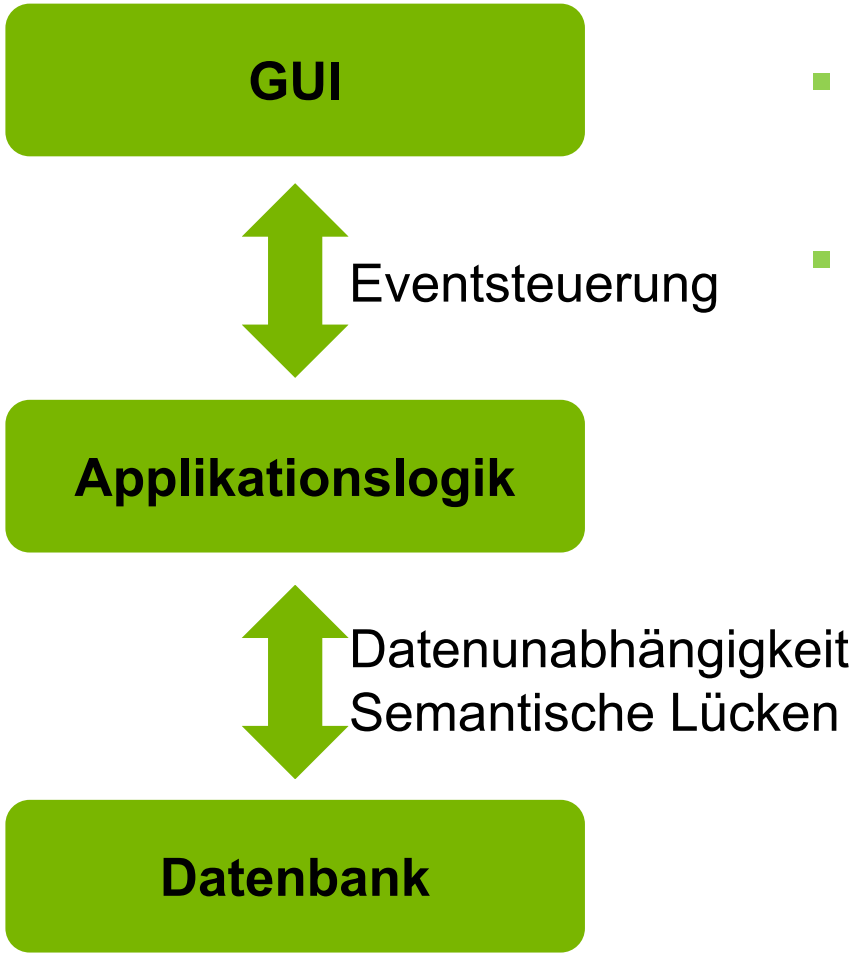
C = Applikation programmieren



Konsistenzgrad in Abhängigkeit des Programmieraufwands



7 Probleme bei der Einbindung in bestehende Systeme



- Eventsteuerung
  - Wie kann GUI auf DB-Änderungen reagieren?
- Geringe Datenunabhängigkeit
  - Bindung von DB-Typen/Strukturen an Programmtypen/strukturen
- Semantische Lücke
  - Abbild des realen Systems in formale Repräsentationen
  - Verschiedene Repräsentationen
    - Typsystems (DB, Programm) verschieden, dadurch u.U. verlustbehaftete Typenumwandlung
    - Auswertungsstrategie Mengen (DB) versus Records, dadurch u.U. aufwendige iterative Bearbeitung Daten notwendig
    - Strukturprimitive unterschiedliche Konzepte für Datenstrukturen (z.B. Zeiger vs. Schlüsselvererbung)

7 Zugriffstechniken

- Zugriffstechniken der Programmiersprachen können variieren

Statement Level Interface (SLI)

- SQL Anweisungen eingebettet in Programmcode
- Verarbeitung durch Präprozessor
- Beispiele: eSQL/C, SQLJ

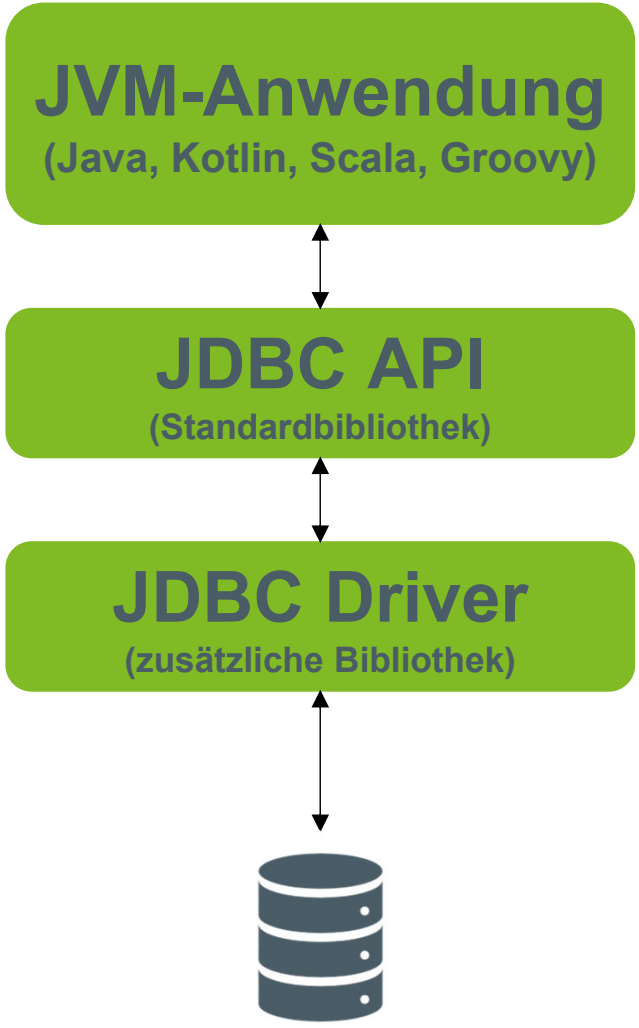
Call Level Interface (CLI)

- Bibliothek mit Zugriffsfunktionen
- Anwendungsprogramm stellt einen Datenbankbenutzer dar
- Beispiele: JDBC (JVM), ADO.NET (C#), ODBC (C, C++), NodeJS-Bibliotheken

7 Beispiel für CLI: JDBC

- Verwendung von SQL als String
- Gleichzeitiger Zugriff auf mehrere Datenbanken
- JDBC ist einfach zu lernen, schwer zu meistern
- „Adaptives“ Programmiermodell
- Grundlage für andere Techniken, wie:

  SQLJ, JPA, Exposed
- Ziele:
  - Einfachheit
  - Robustheit
  - Verfügbarkeit
  - Skalierbarkeit



Allgemeines Vorgehen mit JDBC


```
try {
// 1. Verbindung herstellen
Connection con = DriverManager
.getConnection(
    "jdbc:postgresql://localhost/rezepte",
    "benutzername", "passwort„
);
// 2. Statement erzeugen und ausführen
Statement stmt = con.createStatement();
// 3. Statement ausführen
ResultSet rs = stmt.executeQuery("select * from rezept");
// 4. Ergebnis verwenden
while (rs.next()) {
    System.out.println(rs.getString("titel"));
}
// 5. Ressourcen freigeben
rs.close(); stmt.close(); con.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

7 Statements in JDBC

Statement

- Code wird als String an DBMS gesendet
- DBMS übersetzt, optimiert und führt aus

PreparedStatement

- Code wird mit Platzhaltern im String an DBMS gesendet
  1. DBMS übersetzt und optimiert
  2. Anweisung kann mehrfach mit wechselnden Parametern ausgeführt werden.

CallableStatement

- Aufruf von Stored Procedures

7 Beispiel für PreparedStatement

- Über eine Suchfunktion werden Rezepttitel eingegeben
- Platzhalter in SQL Statement werden an DBMS gesendet
- DBMS erstellt Zugriffsplan

```
PreparedStatement pstmt = conn.prepareStatement(
  "select * from rezept r where r.titel = ?;"
);
while (true) {
  String rezeptTitel = JOptionPane.showInputDialog(
                                          "Rezeptname eingeben:");
  pstmt.setString(1, rezeptTitel);
  ResultSet rs = pstmt.executeQuery();
  if (rs.next()) {
    System.out.printf("Rezept %s ist von %d und dauert %d min.\n",
       rezeptTitel, rs.getInt("benutzer_id"), rs.getInt("dauer_min"));
  } else {
    System.err.printf("Rezept %s wurde nicht gefunden!\n",
                                                  rezeptTitel);
  }
}
```

7 Vorteile des PreparedStatement

- Schnellere Ausführung gleichartiger Abfragen
- Sauberer Code
  - Statt String-Verkettung werden Variablen eingesetzt
  - Das Statement wird nur an einer Stelle „gebaut“
- Keine SQL-Injection möglich!
  - Mehr hierzu: Nächste Woche

Adaptives Programmieren mit JDBC / Metadaten

- Über Interfaces, kann die Struktur der DB (Schema) abgefragt werden

ResultSetMetaData → Informationen über die Ergebnismenge

- int getColumnCount()
- String getColumnName(int column)
- int getColumnType(int column)

DatabaseMetaData → Informationen über das DBMS und das Schema

- String getDatabaseProductName()
- ResultSet getTables(…)
- ResultSet getColumns(…, String tableName)
- String getTimeDateFunctions()
- … und sehr viele mehr.

7 Datentypen von SQL → JDBC

SQL Typ	Java Typ
INT, INTEGER, SERIAL	int
VARCHAR	String
CHAR	String
TEXT	String
DECIMAL, NUMERIC	java.math.BigDecimal
FLOAT, DOUBLE	double
DATE	java.sql.Date
TIME	java.sql.Time
TIMESTAMP	java.sql.Timestamp
BLOB	byte[]

Datenmanipulation mit JDBC

Einfachstes Vorgehen: executeUpdate()

- UPDATE / INSERT wird als (Prepared)Statement mit executeUpdate() abgesetzt
- executeUpdate() liefert dann kein ResultSet, sondern die Anzahl der veränderten Zeilen

Mächtigeres Vorgehen: ResultSet verwenden

- Mit JDBC 2.0 verfügt ein ResultSet über eine updateXXX()-Methode für jeden Datentyp
- Beispiel:

```
rs.updateString(1, "mystring");
rs.updateFloat(2, 10000.0f);
```

7 JDBC – ResultSet – Beispiel

```
statement = conn.createStatement(
  ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
rs = statement.executeQuery("select * from rezept;");
```

Lesen oder Schreiben:

TYPE_FORWARD_ONLY

Kann nur vorwärts durchlaufen werden (Schnappschuss / „Live Update“)

TYPE_SCROLL_INSENSITIVE

Vorwärts und Rückwärts

Zeigt nur die Daten, die vorhanden waren beim Absenden des Statements

TYPE_SCROLL_SENSITIVE

Vorwärts und Rückwärts

Es werden Daten zwischendurch erfolgter Updates sichtbar

Navigation:

- previous() / next(): Springt zur vorherigen /nächsten Zeile
- First() / Last()
- beforeFirst() / afterLast()
- relative(int rows) / absolute(int r)

JDBC - ResultSet

Datenzugriff auf RS ist immer auf eine Zeile der Ergebnismenge.

Werte lesen

- `rs.getString(1)`
- `rs.getString("titel")`

Werte ändern

- `rs.updateBoolean("privat", true)`
- `rs.updateRow()`

Datensätze einfügen

- `rs.moveToInsertRow()`
- `rs.updateString("titel", "Cheeseburger mit Käse")`
- …
- `rs.insertRow()`

Transaktionen mit JDBC

- Transaktionsmodus liegt an der jeweiligen JDBC-Verbindung / Session
- Einfach: Auto-Commit-Modus
  - Jede einzelne SQL-Anweisung führt zu einem Commit
  - `conn.setAutoCommit(false);`
- Nun kann man das Transaktionsende selbst bestimmen!
  - `conn.commit();`
  - `conn.rollback();`

7 Isolationslevel in JDBC

- TRANSACTION_NONE
- TRANSACTION_READ_UNCOMMITTED
- TRANSACTION_READ_COMMITTED (Default in JDBC)
- TRANSACTION_REPEATABLE_READ
- TRANSACTION_SERIALIZABLE

Beispiel:

```
DatabaseMetaData d = con.getMetaData();
if (d.supportsTransactionIsolationLevel(TRANSACTION_SERIALIZABLE) {
  Connection.setTransactionIsolation(TRANSACTION_SERIALIZABLE);
}
```

7 Übung

1. Setzen Sie ein neues Projekt mit Java oder Kotlin auf.
2. Finden Sie einen Weg, den aktuellen JDBC-Treiber für Postgres von *jdbc.postgresql.org* in Ihr Projekt einzubinden.
3. Legen Sie eine neue Datenbank im Playground des FBS an und importieren Sie aus den Vorlagen die Rezepte.
4. Fragen Sie alle Rezepte ab und geben Sie den Benutzernamen des Erstellers aus.
5. Geben Sie alle Benutzer aus, die ein Rezept mit 4 oder 5 Sternen haben.
6. Erstellen Sie ein kleines Interface (grafisch oder Kommandozeile) mit dem der Benutzer aus einer Liste von Zutaten, welche auswählen kann und Sie geben dann die Rezepte aus, die alle ausgewählten Zutaten enthalten.

7 Kontrollfragen

- Welche Sichtweisen auf ein Informationssystem existieren?
- Erläutern Sie die semantische Lücke zwischen Datenbank und Programmiersprache.
- Geben Sie an, was zum Datenbankentwurf gehört.
- Welche Inhalte weisen ein Lasten- bzw. Pflichtenheft bei der DB-Entwicklung auf?
- Wie verhält es sich mit dem Programmieraufwand?
- Welche Probleme gibt es beim Einbinden von Datenbanken in Anwendungsprogramme?
- Welche Datenbankzugriffstechniken kennen Sie?
- Was ist das ResultSet in JDBC? Wofür ist es gut?
- Auf welche Arten können wir Daten mit JDBC updaten?
- Wie können wir in JDBC Transaktionen nutzen?

8 Datenbanken