

# Standards und Infrastrukturen für den Austausch von RFID-Daten

Je Lan Ong



# Mein Assistent: Billy



# Übersicht

- RFID / EPC - Technologie
- Standards und Infrastrukturen (EPCglobal)
- Problematik / Lösungen (BRIDGE)
- Offene Fragen
- Diskussion

# RFID / EPC - Technologie

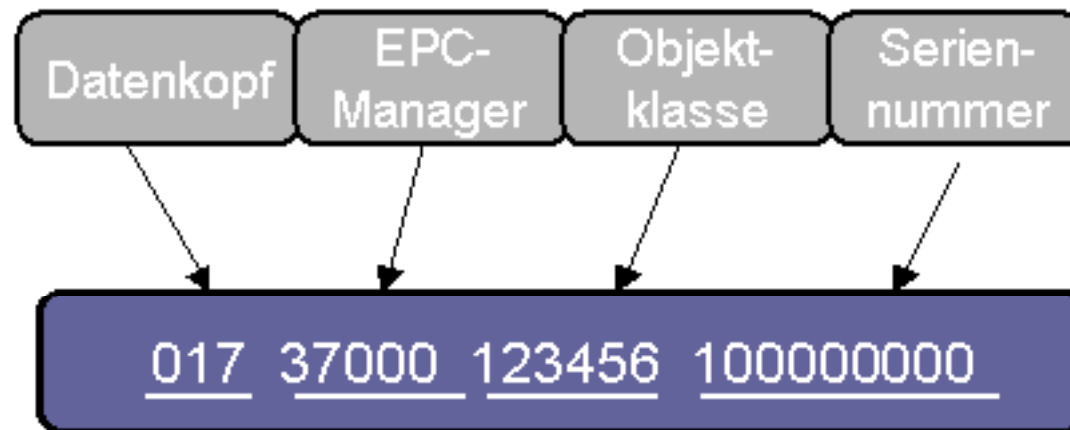
- RFID = **R**adio **F**requency **I**Dentification
- Objekte ausgestattet mit RFID-Chips (Transpondern)
- Lesegeräte erfassen berührungslos die Objekte mittels elektromagnetischer Wellen
- RFID-Chip stellt im wesentlichen den Datenspeicher dar

# RFID / EPC - Technologie

- EPC = Electronic Productcode
- Eindeutige Kennzeichnung
- Besteht aus einer Ziffernfolge
- Unterteilt in Datenkopf, EPC-Manager, Objektklasse und Seriennummer

# RFID / EPC - Technologie

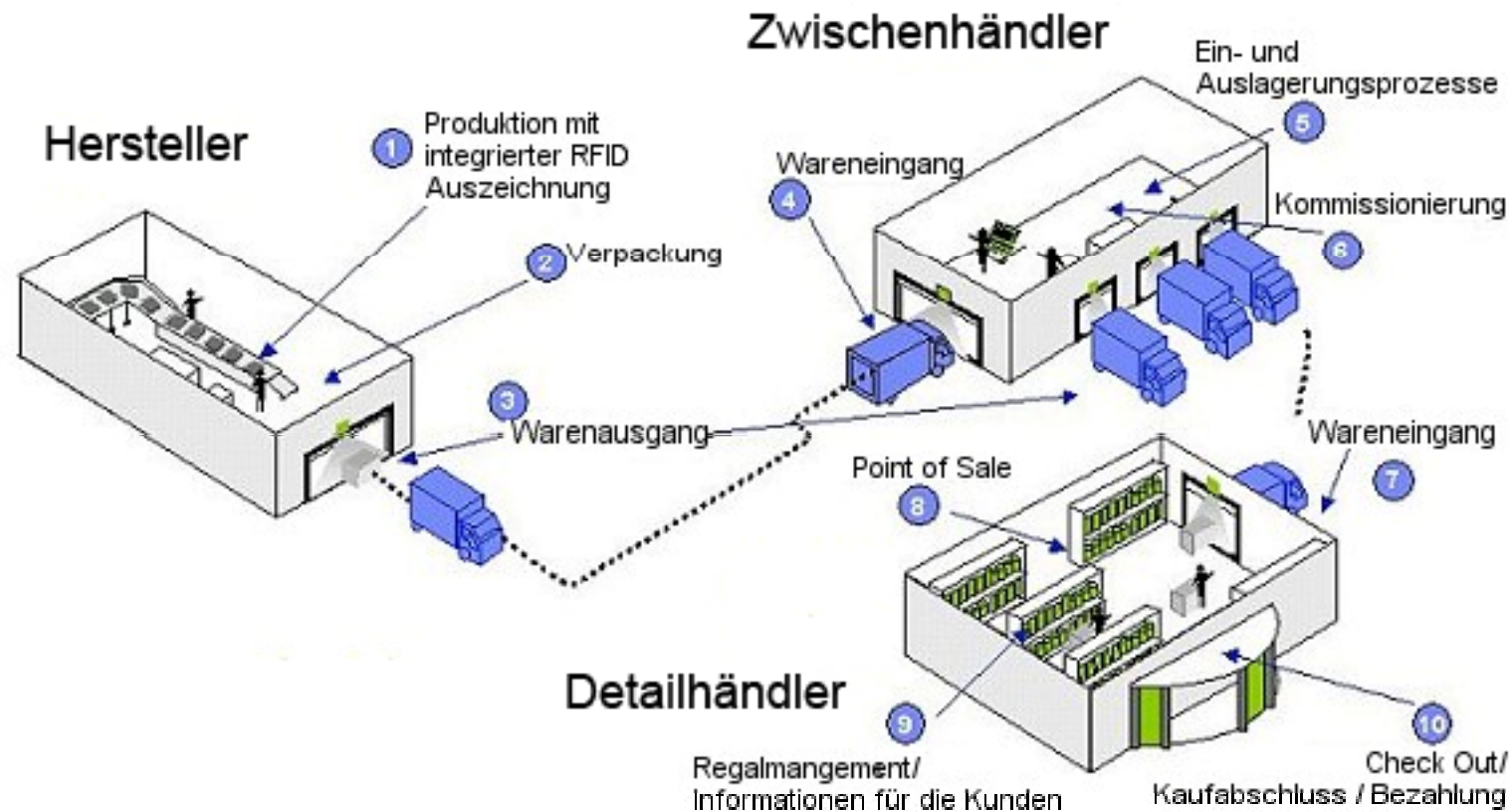
## EPC



# RFID / EPC - Technologie



# RFID / EPC - Technologie





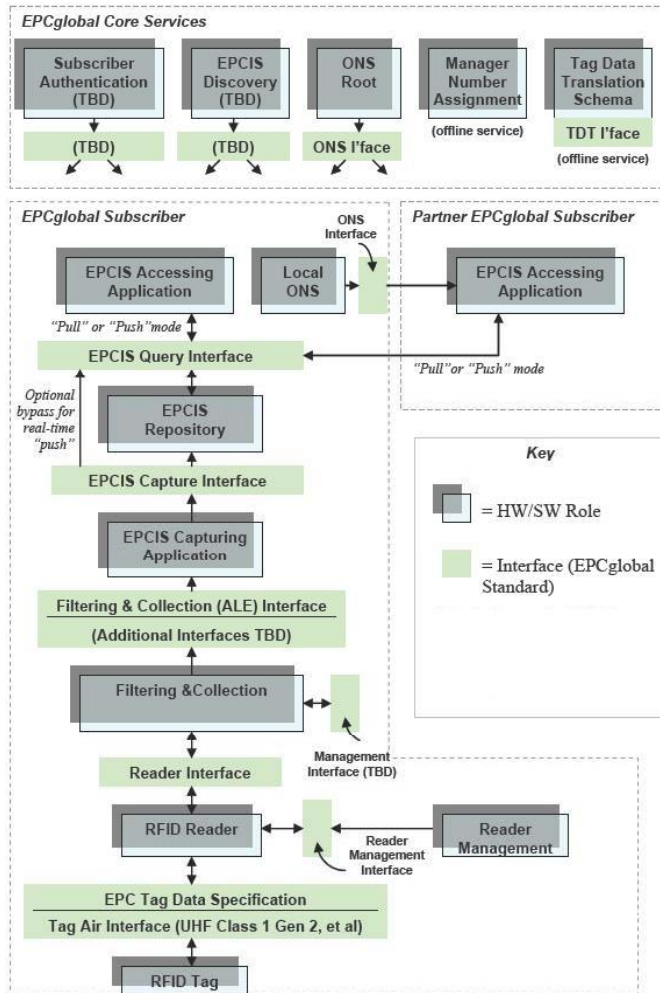
# RFID / EPC - Technologie

- Einsatz von RFID/EPC vereinfacht die Logistik in Wertschöpfungsketten („Supply Chain“)
- Waren werden zwischen Handelspartnern ausgetauscht
- Jeder Handelspartner speichert betriebsinterne Informationen über Waren
- => Erfordert einen Standard

# Standards und Infrastrukturen

- EPC ist von EPCglobal standardisiert
- EPCglobal ist im Jahre 2003 gegründet
- Eine Organisation von Firmen
- Hauptkomponente: „EPCglobal Network“
- Dient dem Austausch von EPC-Informationen über das Internet („Internet der Dinge“)

# Standards und Infrastrukturen

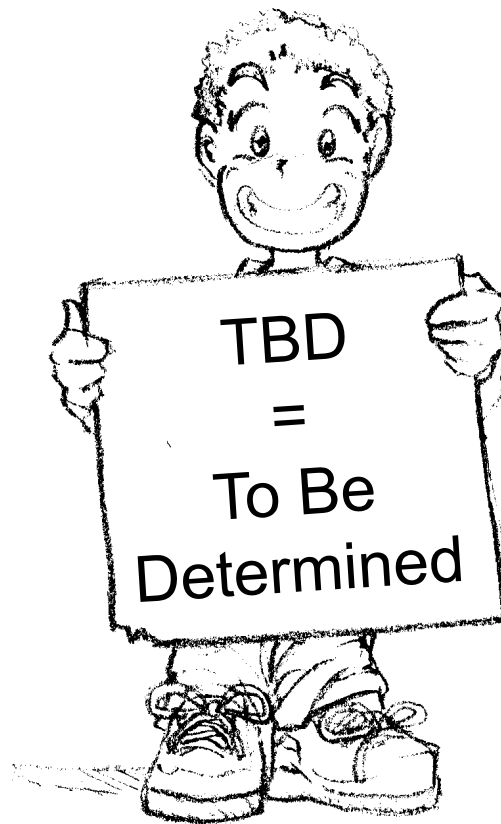


## EPCglobal Architecture

### Framework:

- Object Naming Service (ONS)
- EPC Information Services (EPCIS)
- EPC Discovery Services
- EPC Security Services

# Standards und Infrastrukturen



# Standards und Infrastrukturen

- Jeder Handelspartner speichert EPC-Informationen („Events“) in seinem eigenen EPCIS-Repository
- Folglich ist die gesamten Geschäftsinformationen verteilt
- Ziel: Gesamte Information effizient, sicher und kontrolliert zugänglich zu machen

# Standards und Infrastrukturen

## Beispiel eines Events:

Field	Type	Description
eventTime recordTime eventTimeZoneOffset	(Inherited from EPCISEvent; see Section 7.2.8)	
epcList	List<EPC>	An unordered list of one or more EPCs naming the physical objects to which the event pertained. Each element of this list SHALL be a URI [RFC2396] denoting the unique identity for a physical object. When the unique identity is an Electronic Product Code, the list element SHALL be the "pure identity" URI for the EPC as specified in [TDS1.3], Section 4.1. Implementations MAY accept URI-formatted identifiers other than EPCs.
action	Action	How this event relates to the lifecycle of the EPCs named in this event. See above for more detail.
bizStep	BusinessStepID	(Optional) The business step of which this event was a part.

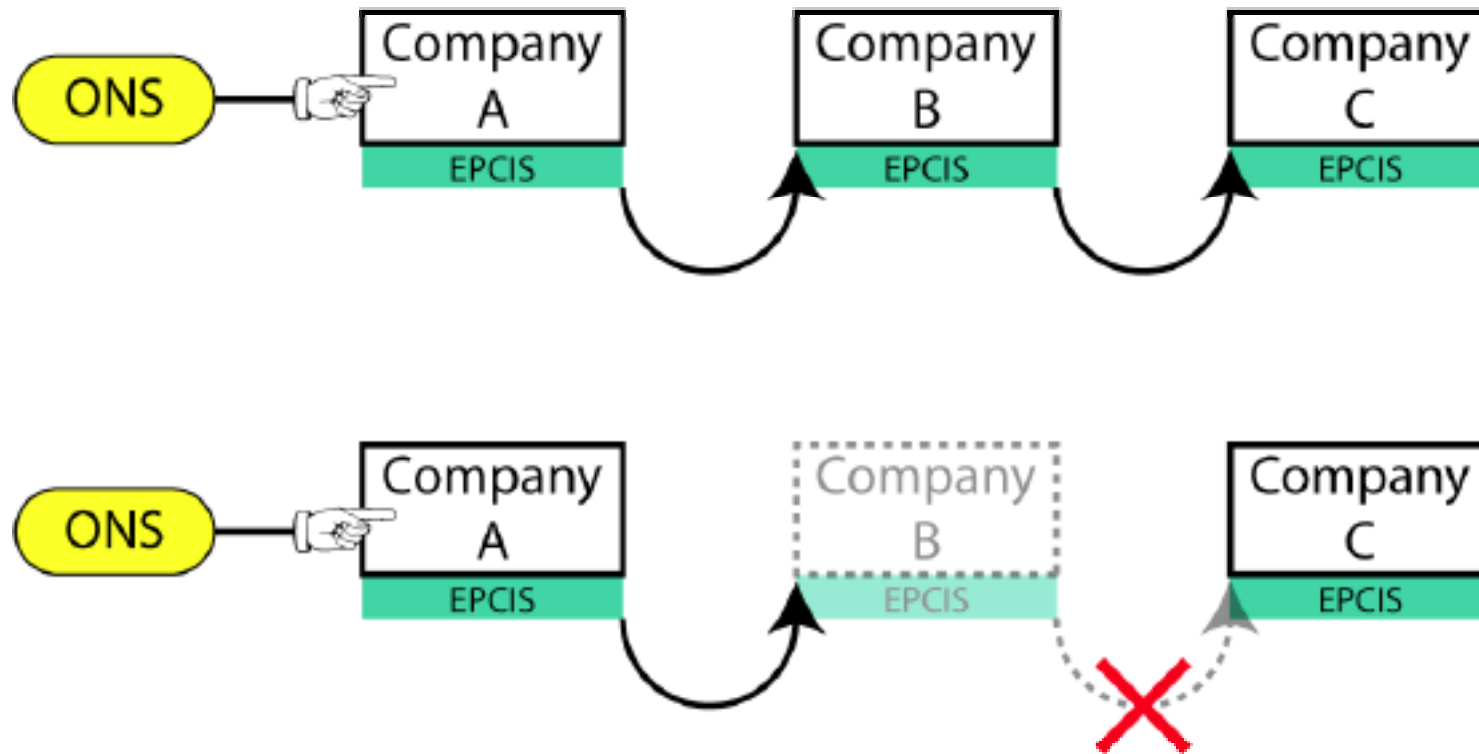
Field	Type	Description
eventTime recordTime eventTimeZoneOffset	(Inherited from EPCISEvent; see Section 7.2.8)	
epcClass	EPClass	The identifier specifying the object class to which the event pertains.
quantity	Int	The quantity of object within the class described by this event.

# Standards und Infrastrukturen

Der Ist-Zustand:

- ONS verwenden um Hersteller ausfindig zu machen
- Beginnend beim Hersteller die Wertschöpfungskette Schritt für Schritt verfolgen
- „Daisy-chain approach“  
(Gänseblümchenverfahren)

# Standards und Infrastrukturen



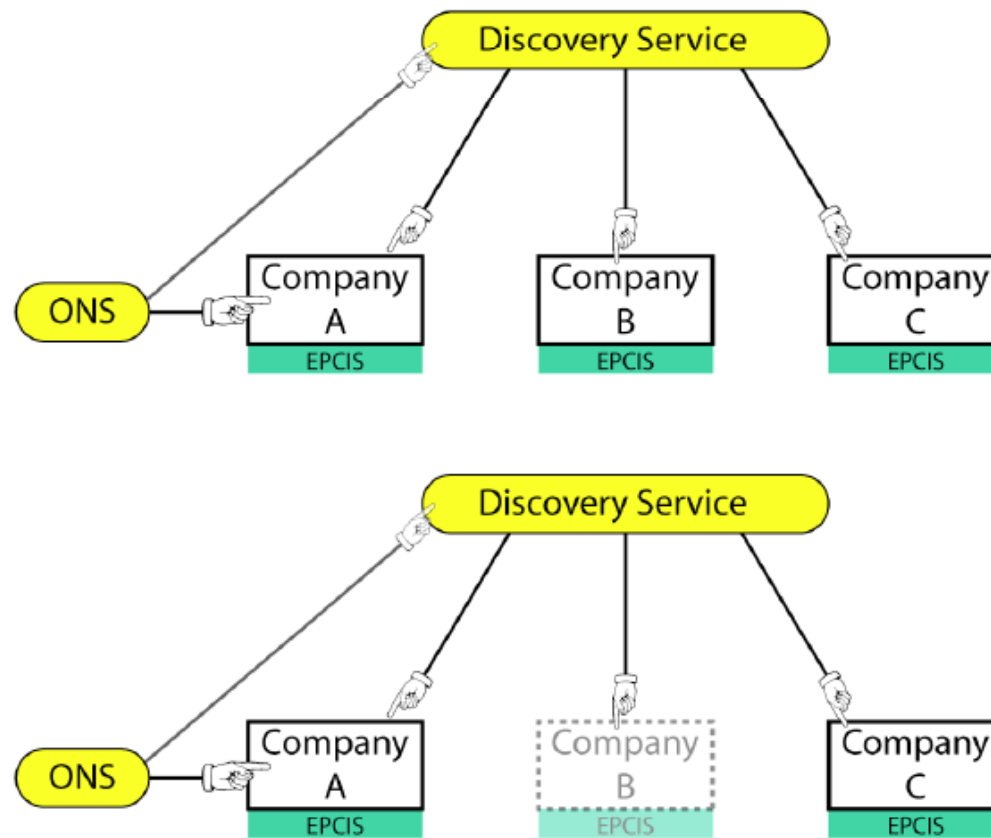


# Standards und Infrastrukturen

Der Soll-Zustand:

- Discovery-Service (EPCIS Discovery) geplant
- Lokalisierung der EPCIS-Repository
- Eingabe: EPC-Nummer
- Ausgabe: Referenz zum EPCIS-Repository
- „Directory Approach“ (Verzeichnisverfahren)

# Standards und Infrastrukturen



# BRIDGE - High level design for Discovery Services

- BRIDGE steht für „**B**uilding **R**adio Frequency **I**dentification for the **G**lobal **E**nvironment“
- Projekt gestartet in 2006 (Dauert 3 Jahre)
- Verband bestehend aus Unternehmen, Universitäten etc. aus Europa und China
- Ziel: Technische Lösungen zu entwickeln im Bereich RFID/EPC-Technologien

# BRIDGE - High level design for Discovery Services

Zusammenspiel zwischen:

- Den Klienten („Clients“) – Interessiert an Daten in EPCIS-Repositories
- Dem Discovery Service („Der Vermittler“)
- Den EPCIS-Repositories

# BRIDGE - High level design for Discovery Services

Das Kommunikationsmodell besteht aus:

- Setup-Phase: Initialisierung mit dem „Discovery-Service“
- Discovery-Phase: Gegenseitiges Finden zwischen Clients und EPCIS-Repositories
- Service-Phase: Eigentlicher Datenaustausch zwischen Clients und EPCIS-Repositories

# BRIDGE - High level design for Discovery Services

Mögliche Datensätze beim Discovery-Service:

- Tupeln der Form (EPC, EPCIS-Referenz).
- Vollständig replizierte Daten der EPCIS-Repositories
- Einfache Anfragen: EPC-Nummern
- Vollständige Anfragen: EPC-Nummern+ Parametern (z.B. Anzahl Suchresultate)

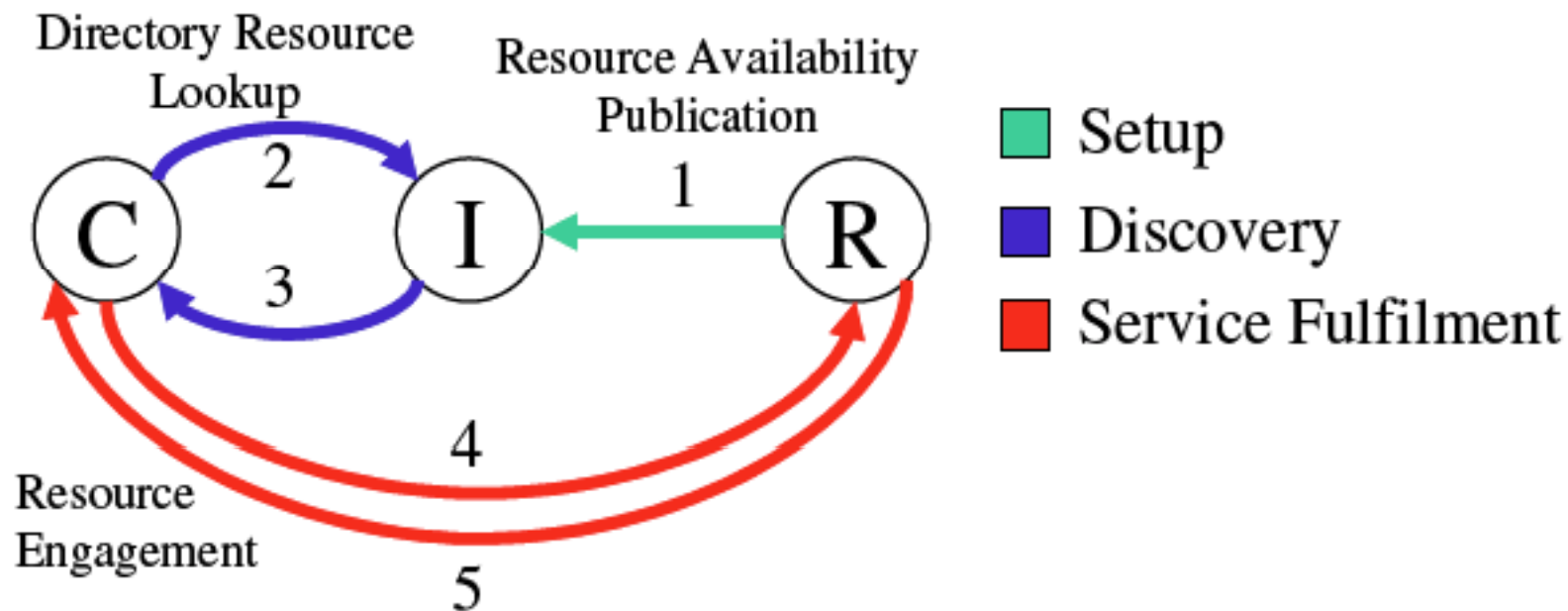
# BRIDGE - High level design for Discovery Services

Zwei verschiedene Arten von Queries:

- One-Off-Queries: Einmalige Anfragen um bestehende Daten abzufragen.
- Standing queries: Bestehende Anfragen um zukünftige Ereignisse informiert zu werden.

# BRIDGE - High level design for Discovery Services

## Directory of Resources





# BRIDGE - High level design for Discovery Services

## Directory of Resources

### Vorteile:

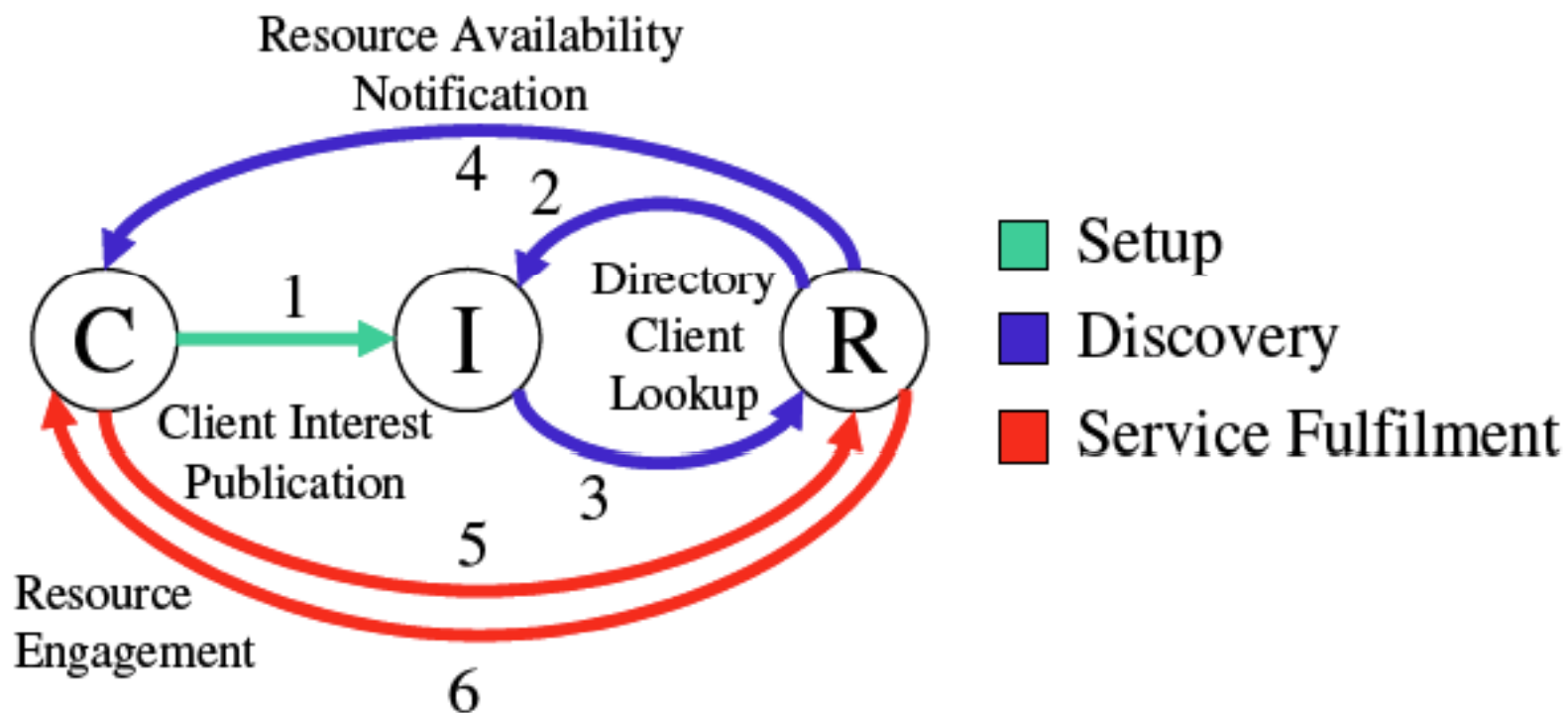
- Klient hat komplette Liste von Referenzen
- Zugriffskontrolle bei den Ressourcen

### Nachteile:

- Sicherheit (Sichtbarkeit der EPCIS-Repositories)
- EPC-Nummer als Schlüssel für Speicherort

# BRIDGE - High level design for Discovery Services

## Directory of Clients



# BRIDGE - High level design for Discovery Services

Directory of Clients

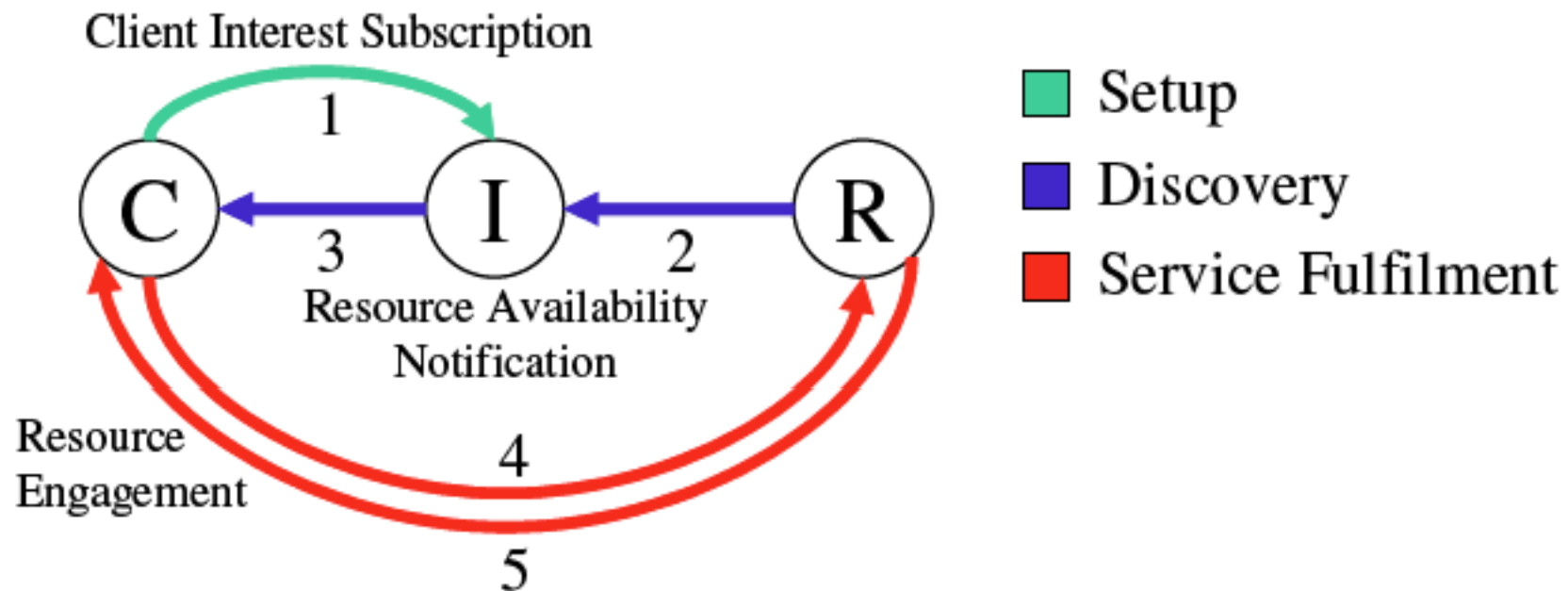
Nachteile:

- Polling der EPCIS-Repositories ineffizient
- Client hat keine Kontrolle über seine Anfragen

=> Modell ungeeignet

# BRIDGE - High level design for Discovery Services

## Notification of Resources



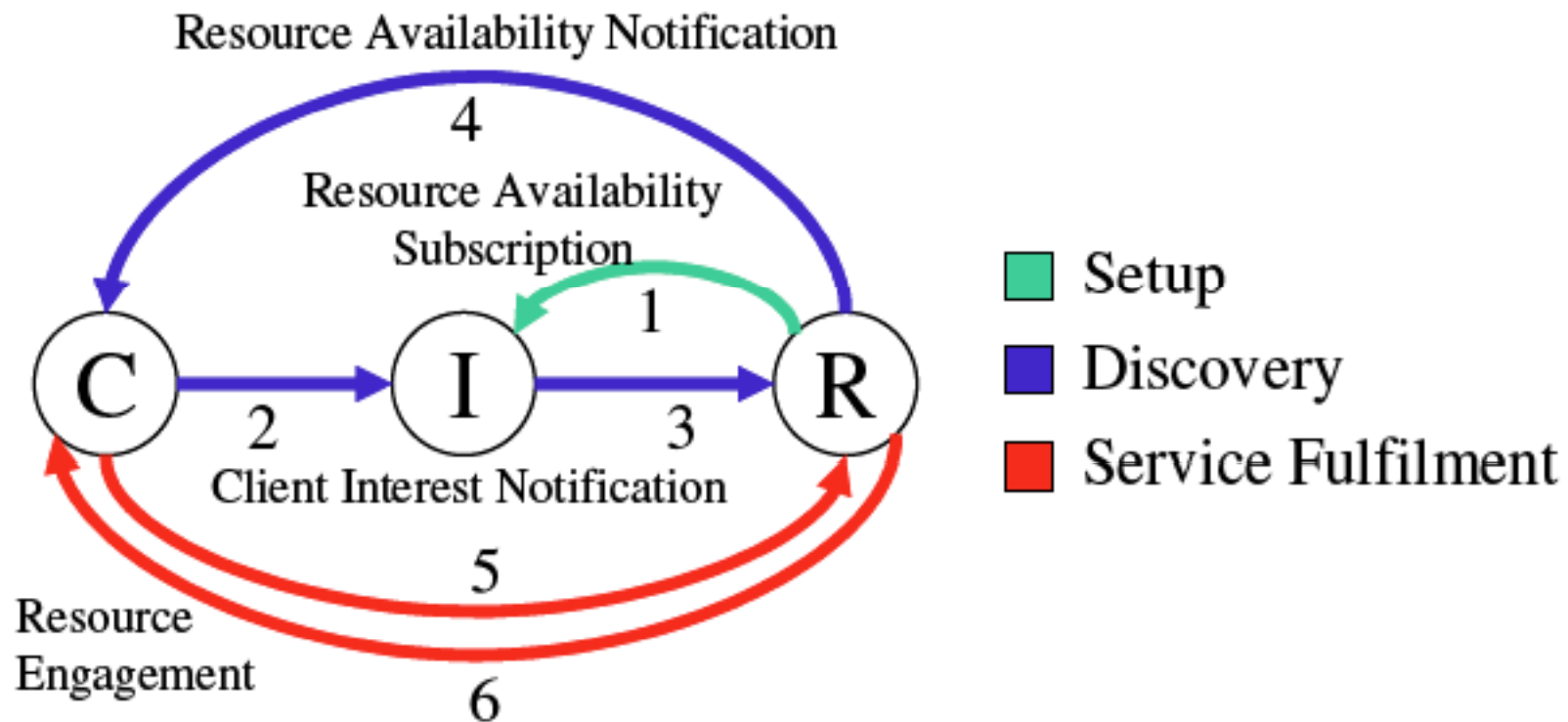
# BRIDGE - High level design for Discovery Services

## Notification of Resources

- Prinzipiell wie das Modell „Directory of Resources“
- Verwendet das „Publish/Subscribe“-Paradigma
- Vorteil: Effizienter als „Polling“
- Nachteil: Sicherheit

# BRIDGE - High level design for Discovery Services

## Notification of Clients



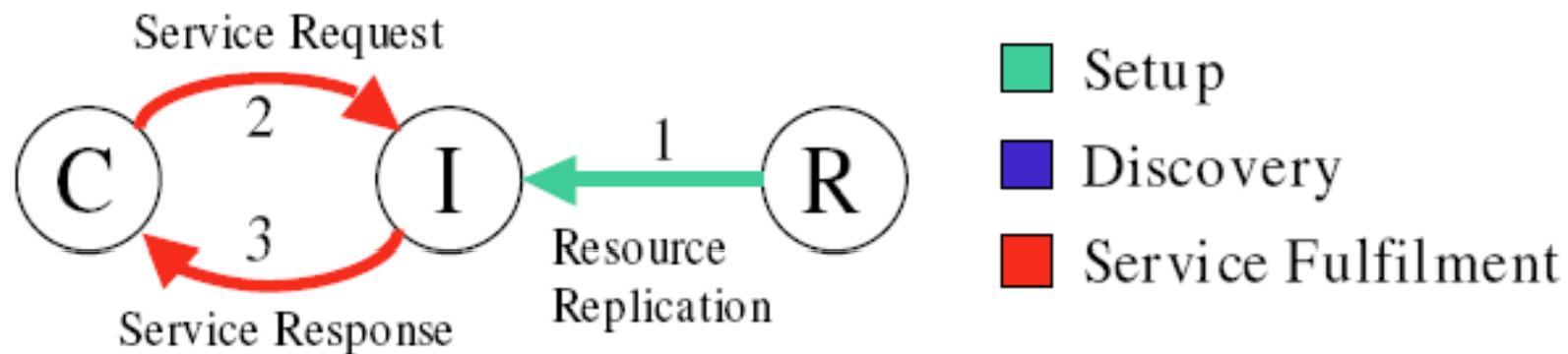
# BRIDGE - High level design for Discovery Services

## Notification of Clients

- Client bekommt nicht „gratis“ die vollständige Liste aller Referenzen (Nachteil für Client)
- EPCIS-Repository entscheidet welcher Client informiert werden soll (Vorteil für EPCIS-Repository)

# BRIDGE - High level design for Discovery Services

## Meta Resource





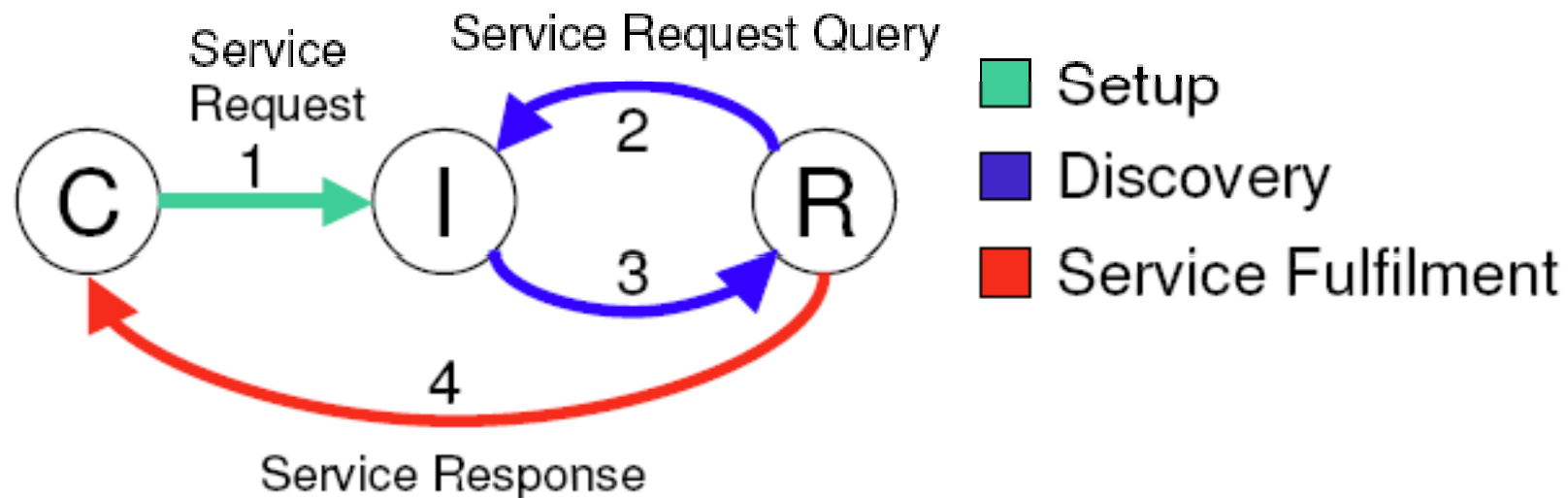
# BRIDGE - High level design for Discovery Services

## Meta Resource

- Vorteil: Clients und EPCIS-Repositories entkoppelt
- Nachteil: Grosse Verantwortung beim Discovery-Service

# BRIDGE - High level design for Discovery Services

## Meta Client



# BRIDGE - High level design for Discovery Services

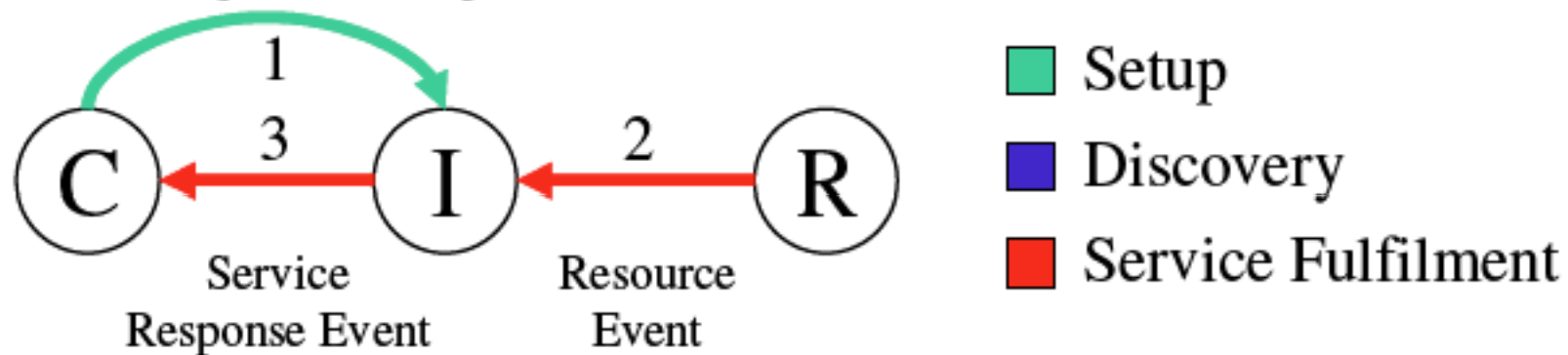
## Meta Client

- Ungeeignet für One-Off-Queries
- Anfragen nicht sofort bearbeitet
- Für Standing-Queries allerdings gut geeignet

# BRIDGE - High level design for Discovery Services

## Notification of Events

Service Request Subscription



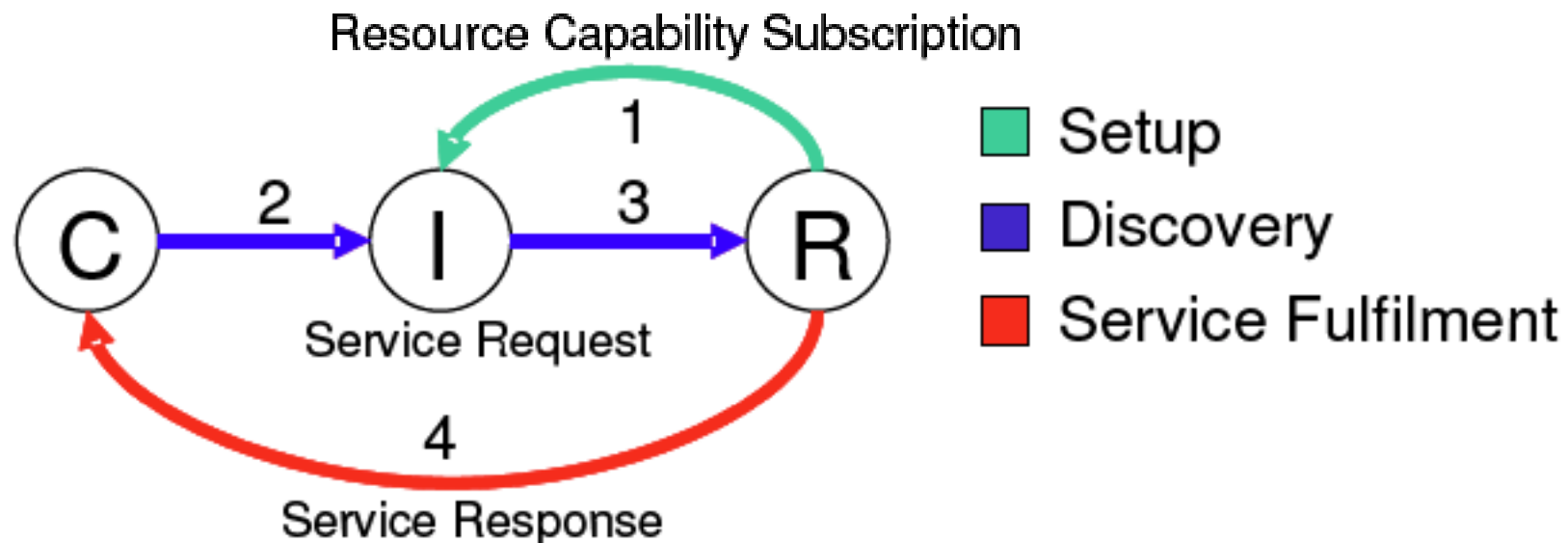
# BRIDGE - High level design for Discovery Services

## Notification of Events

- Effiziente Informationsverteilung
- Nützlich z.B. in der Sensortechnik
- Hier eher ungeeignet: EPCIS-Repositories verlieren Kontrolle über ihre Daten

# BRIDGE - High level design for Discovery Services

## Query Propagation



# BRIDGE - High level design for Discovery Services

## Query Propagation

- „Privatsphäre“ der EPCIS-Repositories geschützt
- Jedes EPCIS-Repository hat volle Kontrolle über die eigenen Daten
- Anfällig auf gezielte Angriffe („Denial-Of-Service Attacks“)

# BRIDGE - High level design for Discovery Services

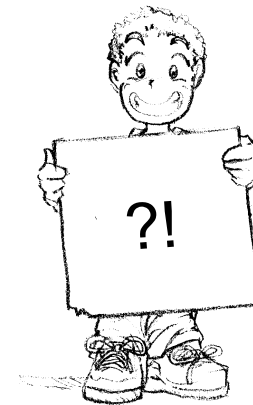
## Schlussbemerkungen:

- Modelle rein aus technischer Sicht betrachtet
- Modelle nicht von Natur aus gut oder schlecht
- Muss abhängig vom Anwendungsfall („Use Case“) beurteilen
- Z.B. Modell beim Paketversand im Internet vs. Paketversand (Gepäck) bei Flughäfen



# Offene Fragen

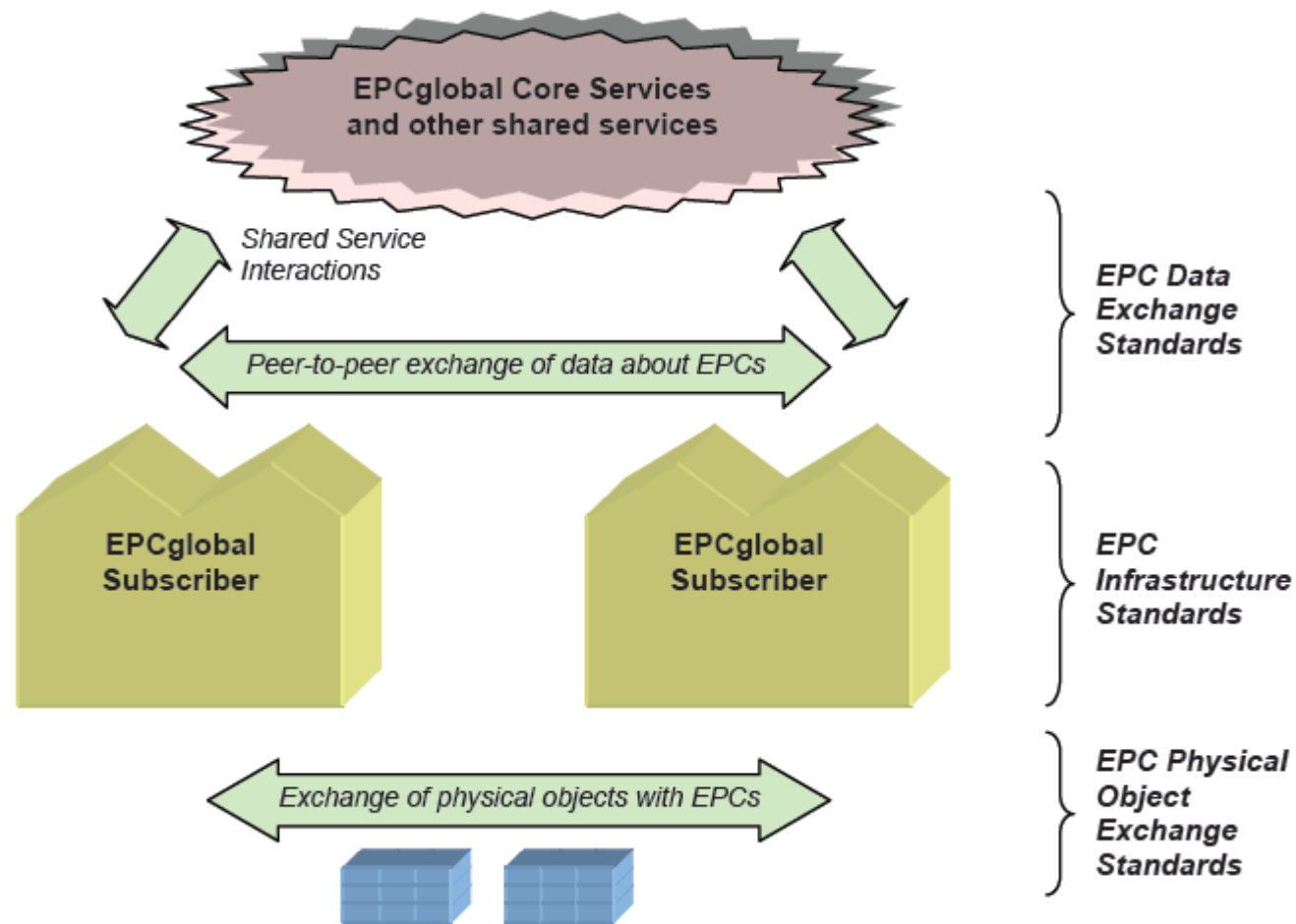
- Wer finanziert das System?
- Wie wird das System eingesetzt? (Deployment)
- Wer hat die Macht? (Kontrolle über das System)
- Privacy: z.B. Spionage der Konkurrenz
- ...



# Diskussion – Anhang 1

Modell	Vertrauen des Klienten	Vertrauen der Ressourcen	Entscheidung
Directory of Resources	Gut	Bedenklich	Geeignet
Directory of Clients	Bedenklich	Gut	Abgelehnt
Notification of Resources	Gut	Bedenklich	Geeignet
Notification of Clients	Bedenklich	Gut	Geeignet
Meta Resource	Gut	Schlecht	Abgelehnt
Meta Client	Bedenklich	Gut	Abgelehnt
Notification of Events	Gut	Schlecht	Abgelehnt
Query Propagation	Bedenklich	Gut	Geeignet

# Diskussion – Anhang 2



## Diskussion – Anhang 3

### EPCIS Data Categories:

- Static Data: Class-level Static Data, Instance-level Static Data
- Transactional Data: Instance Observation (z.B. Shipping dates), Quantity Observations, Business Transaction Observations (z.B. Shipped in purchase order #23)

# Vielen Dank für Ihre Aufmerksamkeit

