# Eine einheitliche, formale Grundlage für dienstbasierte Architekturen

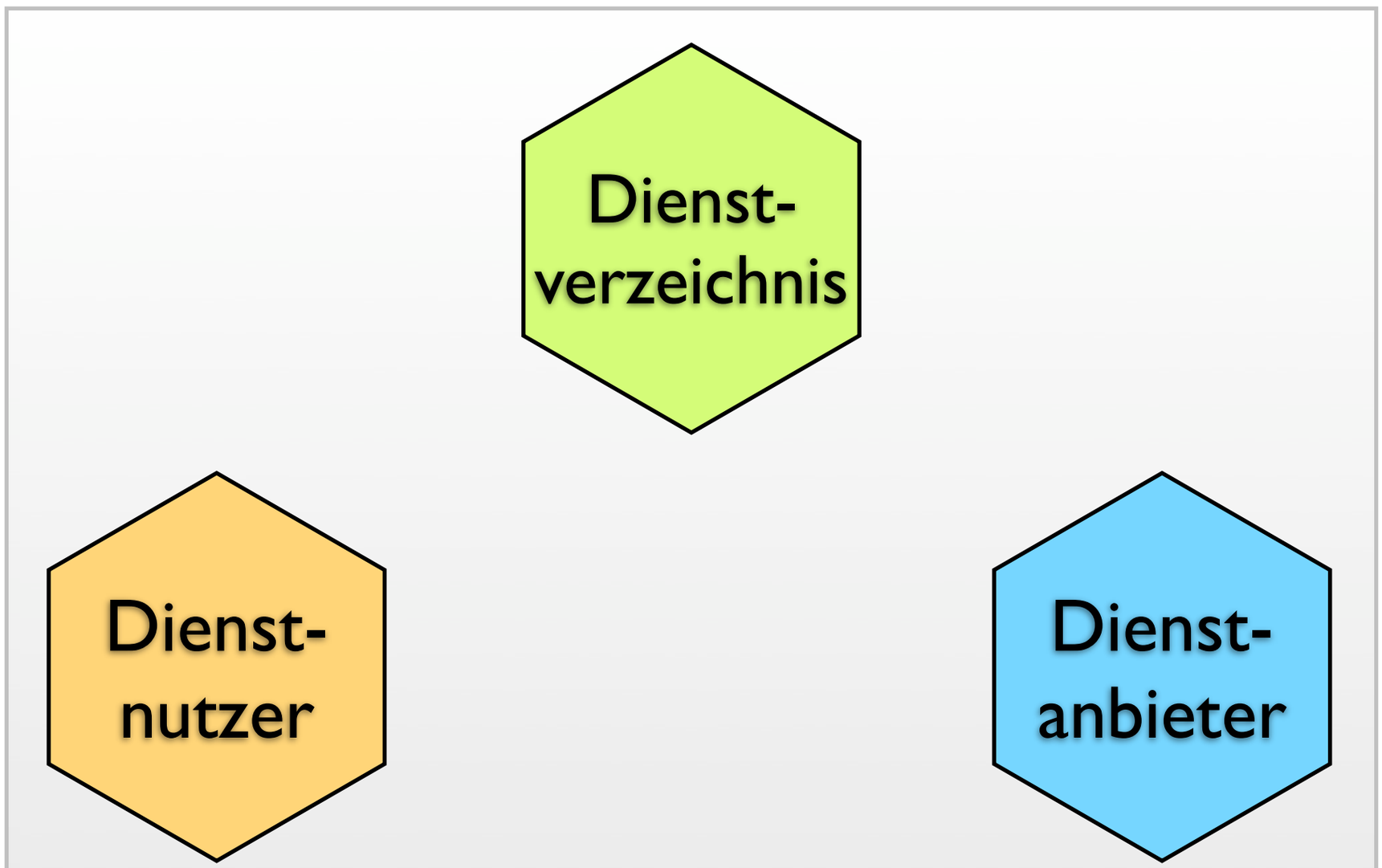
Frank Puhlmann

Business Process Technology
Hasso Plattner Institut
Potsdam, Germany

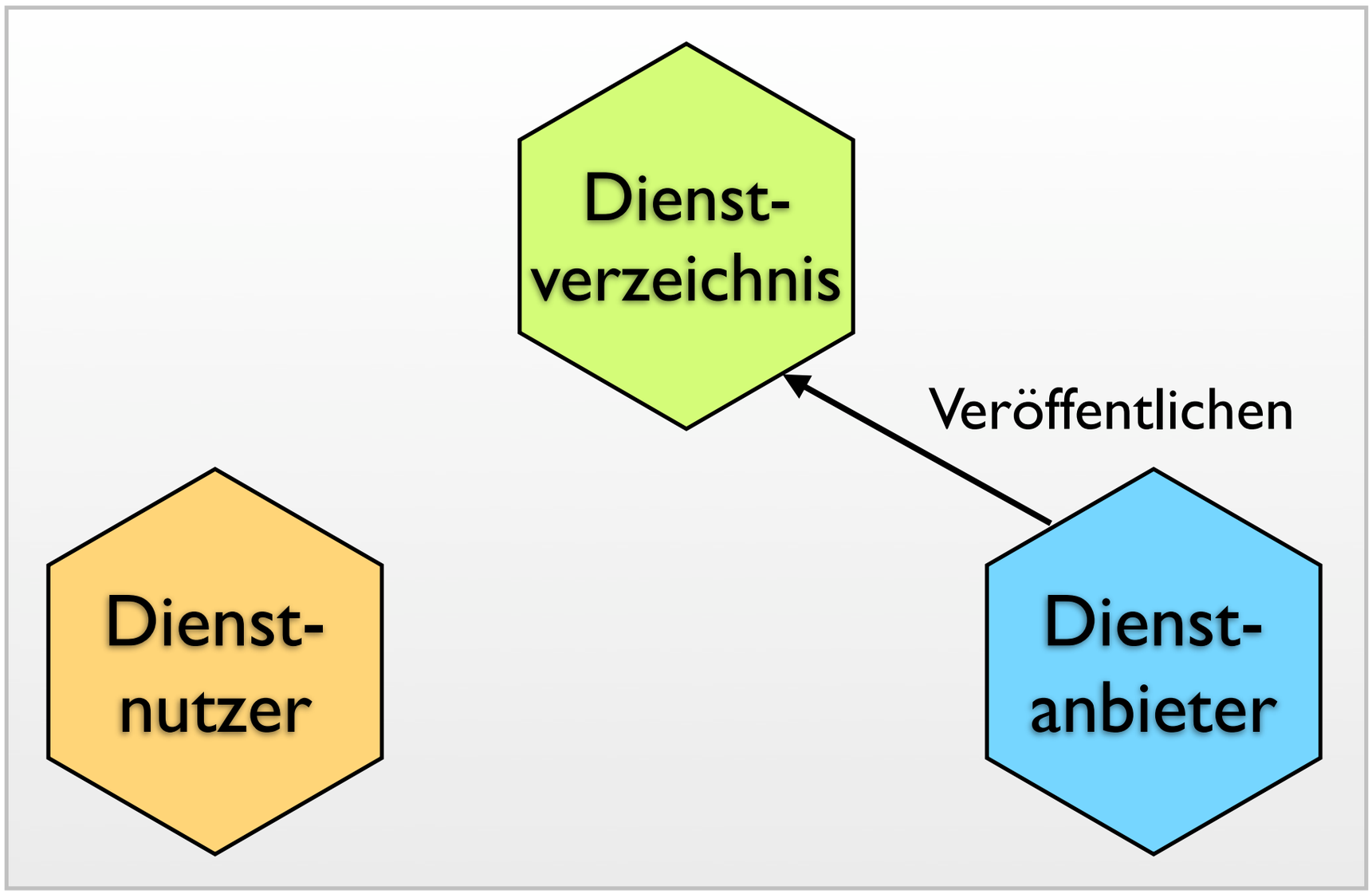
# Gliederung

- Motivation
- Betrachtete Konzepte
  - Interaktionen
  - Prozesse
  - Daten
- Zusammenfassung

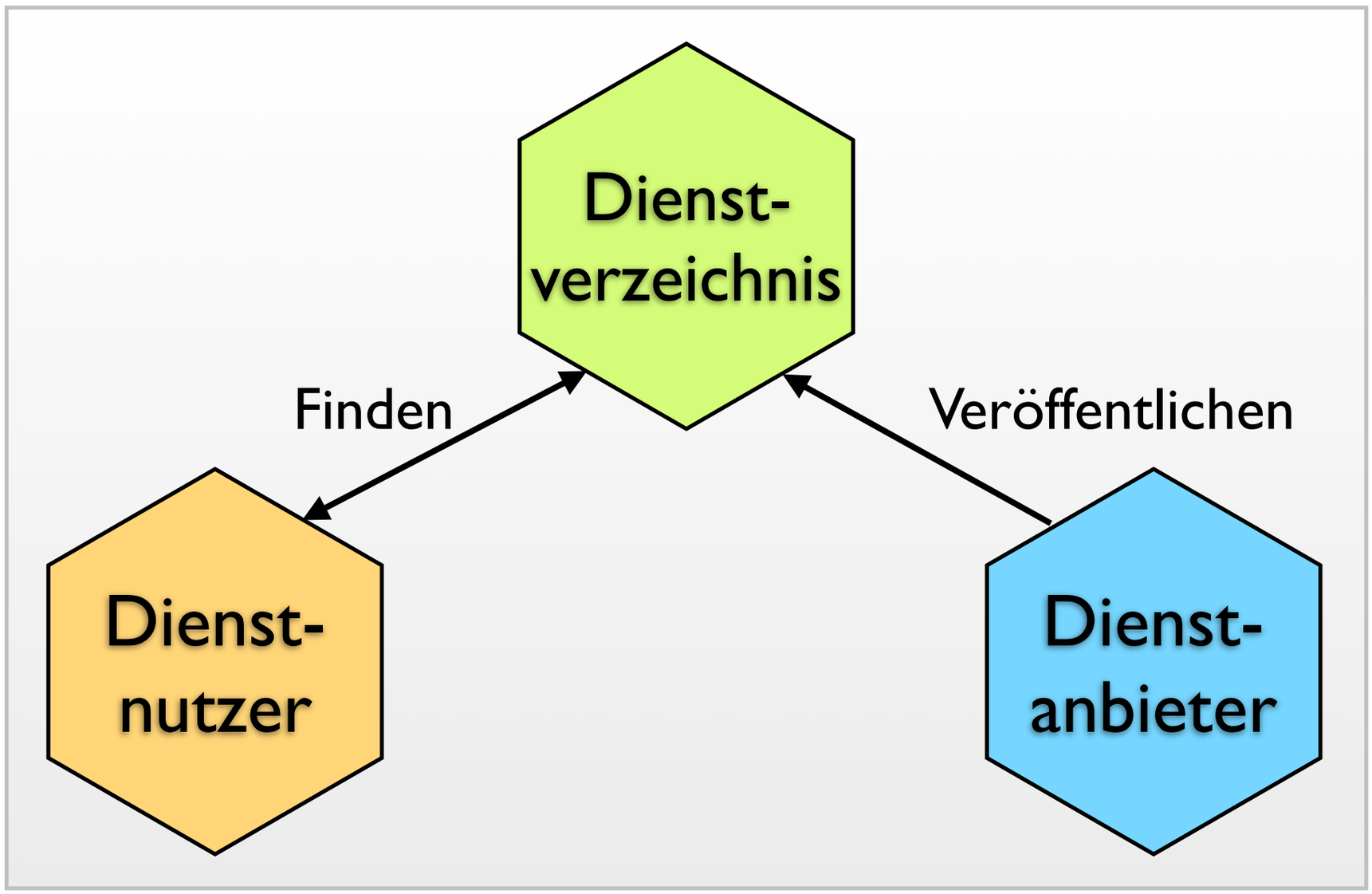
# Motivation



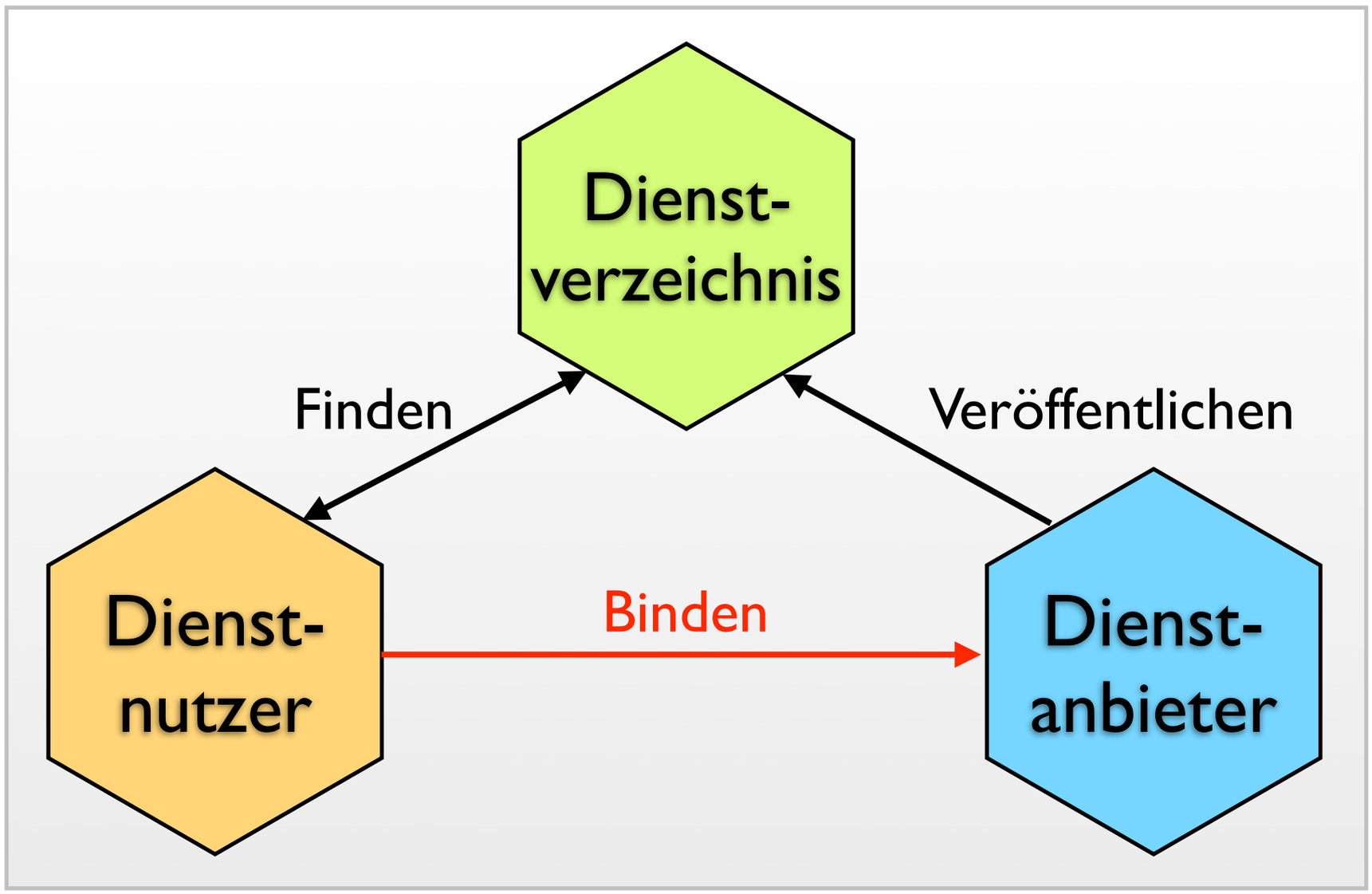

# Dienstbasierte Architektur



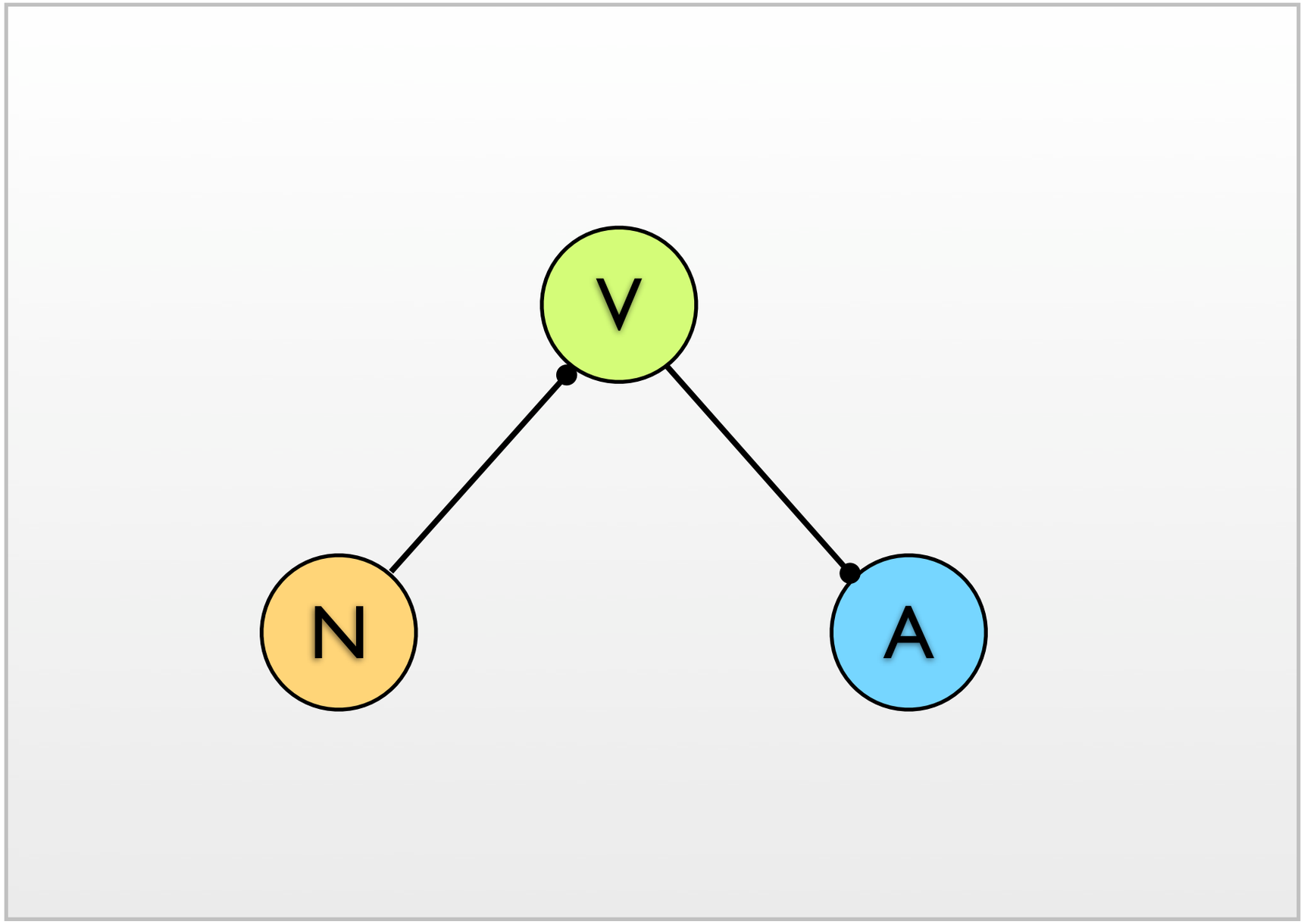

# Dienstbasierte Architektur



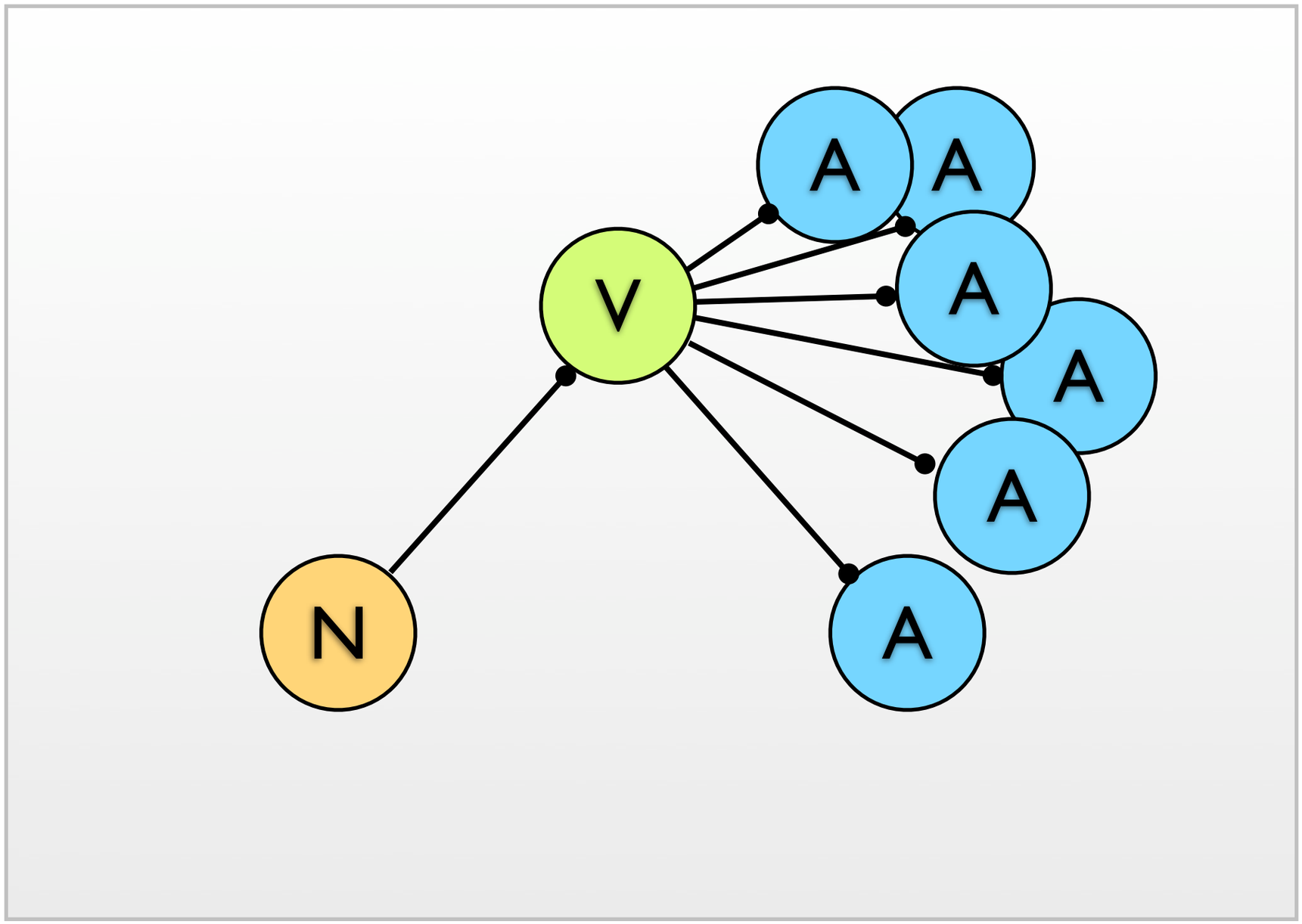

# Dienstbasierte Architektur



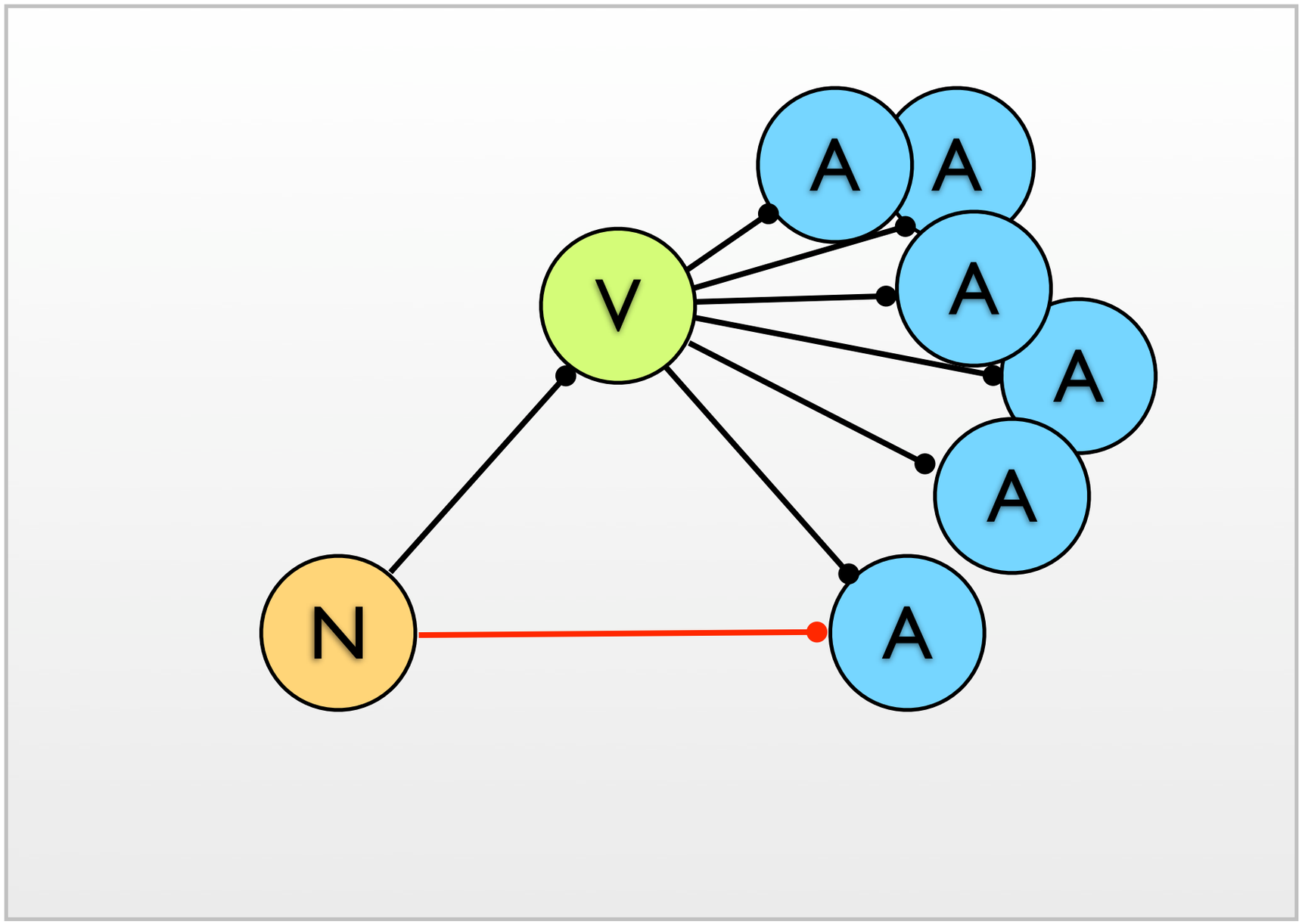

# Dienstbasierte Architektur



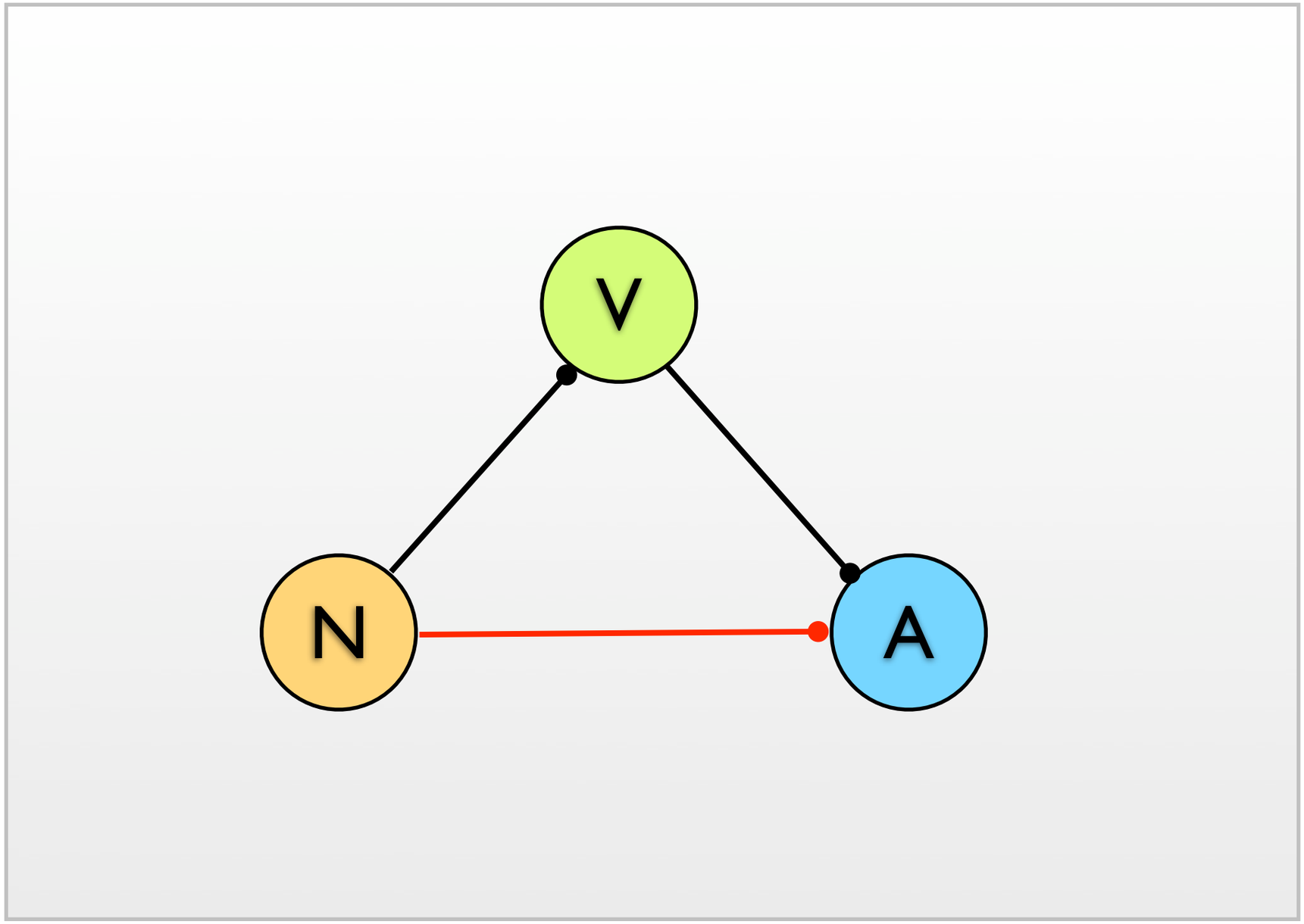

# Pi-Calculus Linkmobilität



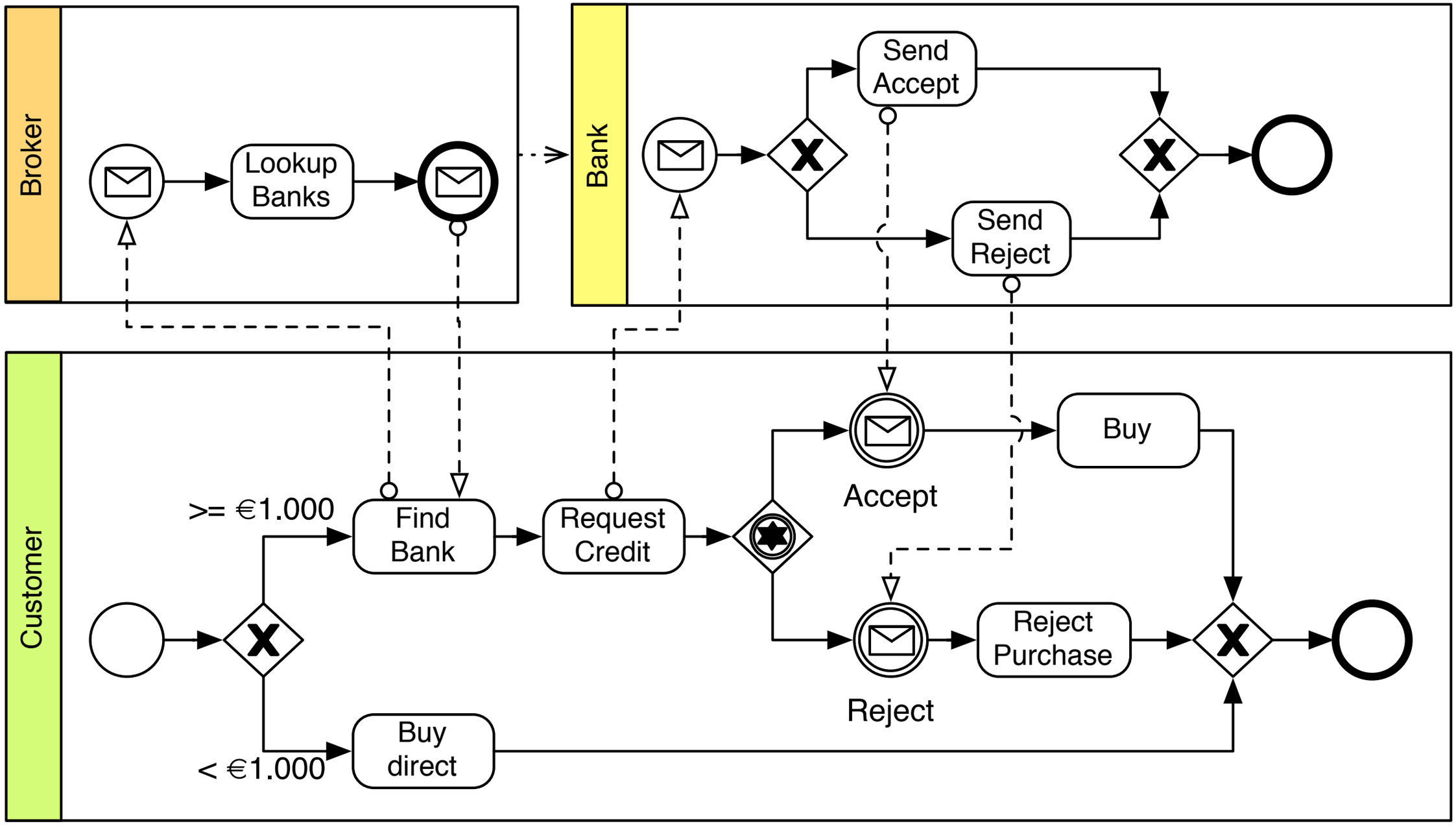

# Pi-Calculus Linkmobilität




# Pi-Calculus Linkmobilität




# Pi-Calculus Linkmobilität




# Dienstbasierte Architektur

# Konzepte

# Der Pi-Calculus

- Eine Prozessalgebra für Systeme mit Linkmobilität
- Syntax in BNF:

$$P ::= M \mid P|P' \mid \mathbf{v}zP \mid !P \mid P(y_1, \cdots, y_n)$$
$$M ::= \mathbf{0} \mid \pi.P \mid M + M'$$
$$\pi ::= \overline{x}\langle\tilde{y}\rangle \mid x(\tilde{z}) \mid \tau \mid [x = y]\pi \ .$$

# Interaktionen

- Dienstaufrufe:

# Interaktionen

- Dienstaufrufe:

$$A = \overline{b}\langle msg \rangle.A' \qquad B = b(msg).B'$$

# Interaktionen

- Dienstaufrufe:

$$A = \overline{b}\langle msg \rangle.A' \qquad B = b(msg).B'$$

Statisches Binden:

$$S = (\mathbf{v}b)(A \mid B)$$

# Interaktionen

- Dienstaufrufe:

$$A = \overline{b}\langle msg \rangle.A' \qquad B = b(msg).B'$$

Statisches Binden:

$$S = (\mathbf{v}b)(A \mid B)$$

Dynamisches Binden:

$$S = (\mathbf{v}lookup)(lookup(b).A \mid ((\mathbf{v}b)B \mid R))$$

# Interaktionen

- Dienstaufrufe:

$$A = \overline{b}\langle msg \rangle.A' \qquad B = b(msg).B'$$

Statisches Binden:

$$S = (\mathbf{v}b)(A \mid B)$$

Dynamisches Binden:

$$S = (\mathbf{v}lookup)(lookup(b).A \mid ((\mathbf{v}b)B \mid R))$$

$$R = \overline{\widetilde{lookup}}\langle b \rangle.R$$

# Prozesse

- Basierend auf Workflow-Pattern-Formalisierungen:

# Prozesse

- Basierend auf Workflow-Pattern-Formalisierungen:

Sequence: $A = \tau_A.\bar{b}.\mathbf{0} \qquad B = b.\tau_B.B'$

# Prozesse

- Basierend auf Workflow-Pattern-Formalisierungen:

Sequence: $A = \tau_A.\bar{b}.\mathbf{0} \qquad B = b.\tau_B.B'$

$$SYSTEM = (\mathbf{v}b)(A \mid B)$$

# Prozesse

- Basierend auf Workflow-Pattern-Formalisierungen:

Sequence: $A = \tau_A.\overline{b}.\mathbf{0} \qquad B = b.\tau_B.B'$

$$SYSTEM = (\mathbf{v}b)(A \mid B)$$

Exclusive Choice: $A = \tau_A.(\overline{b}.\mathbf{0} + \overline{c}.\mathbf{0})$

# Prozesse

- Basierend auf Workflow-Pattern-Formalisierungen:

Sequence: $A = \tau_A.\overline{b}.\mathbf{0} \qquad B = b.\tau_B.B'$

$$SYSTEM = (\mathbf{v}b)(A \mid B)$$

Exclusive Choice: $A = \tau_A.(\overline{b}.\mathbf{0} + \overline{c}.\mathbf{0})$

Simple Merge: $D = d_1.\tau_D.D' + d_2.\tau_D.D'$ .

# Prozesse

- Basierend auf Workflow-Pattern-Formalisierungen:

Sequence: $A = \tau_A.\overline{b}.\mathbf{0} \qquad B = b.\tau_B.B'$

$$SYSTEM = (\mathbf{v}b)(A \mid B)$$

Exclusive Choice: $A = \tau_A.(\overline{b}.\mathbf{0} + \overline{c}.\mathbf{0})$

Simple Merge: $D = d_1.\tau_D.D' + d_2.\tau_D.D' \ .$

Deferred Choice: $A = \tau_A.(b_{env}.\overline{b}.\mathbf{0} + c_{env}.\overline{c}.\mathbf{0})$

# Daten

- Keine native Unterstützung
- Allerdings: Lambda-Calculus im Pi-Calculus darstellbar
- Erweiterungen zur direkten Unterstützung von Daten vorhanden; allerdings
  - Auf Kosten der Beweismöglichkeiten
- Beispiele im Papier

# Zusammenfassung

# Anwendungen

- Grafische Modelierung
  - BPMN nach Pi-Calculus Konvertierer
- Ausführung & Simulation
  - PiVizTool
- Verifikation
  - Verschiedene Soundness-Eigenschaften (MWB, ABC)

# Ergebnis

- Der Pi-Calculus unterstützt die formale Modellierung der betrachteten Konzepte
  - Interaktionen inkl. dynamischen Binden,
  - Prozesse sowie
  - Daten
- in einer absteigenden Ausdrucksfähigkeit.

# Vielen Dank!