

# **AntiPatterns: Stovepipe Enterprise Stovepipe System**

Roland Stigge

stigge@informatik.hu-berlin.de

Humboldt Universität zu Berlin

2. Februar 2002



# Stovepipe Enterprise/System: Überblick

- Hintergrund
- Symptome
- Ausnahmen
- Lösung
- Verwandte AntiPatterns

# Stovepipe Enterprise



# Hintergrund

- (dt.) “Ofenrohr-Unternehmen”
- Bezogen auf Systeme in einem Unternehmen
- AKA: Islands of automation
- Gründe:
  - Eile
  - Apathie
  - Ignoranz

# Symptome

- inkompatible Technologie
- wenig Interoperabilität
  - auch bei gleichen Standards
- keine (wenig) Dokumentation
- geringe Erweiterbarkeit, Wiederverwendbarkeit

## Symptome (2)

- Redundanz
- hohe Verwaltungskosten bei Veränderungen
- Probleme bei Mitarbeiterverlust
- viele Adapter

# Ursachen

- fehlende
  - technologische Unternehmensstrategie
  - Kooperation zw. Entwicklern
  - Kooperation zw. Projekten
  - Kompetenz
  - horizontale Schnittstellen bei Systemintegration

# Ausnahmen

- Übernahme/Fusion von Unternehmen
- gemeinsame Dienste (z.B. DB2 / Oracle)



# Lösung

- Definition & Vereinheitlichung von:
  1. Standards
  2. Betriebsumgebungen (Produkte)
  3. System-Profilen (Verwendung der Produkte)

# Verwandte AntiPatterns

- Reinvent the Wheel
- Hack

# Fragen?



# Stovepipe System



# Hintergrund

- (dt.) “Ofenrohr-System”
- bezogen auf Software-Subsysteme
- Subsysteme sind Hacks
- aka:
  - Ad-Hoc-Integration
  - Legacy System (Altlasten-System)
  - Uncle Sam Special
- auch: “Working around other Hacks”
- Gründe: Eile, Ignoranz, Faulheit

# Symptome

- Mangel an:
  - Konsistenz v. Dokumentation und Implementation
  - Erweiterbarkeit
  - Interoperabilität
- Budget- und Zeitplanüberziehung
- unnötige Komplexität

## Symptome (2)

- unzufriedene Benutzer
- Benutzer brauchen Workarounds
- Komplexe Installationsverfahren
- Point-to-Point-Integration

# Ursachen

- zu viel Infrastruktur
- Mangel an:
  - Subsystem-Abstraktion
  - Architektur
  - Kompetenz
- enge Bindung von Klassen



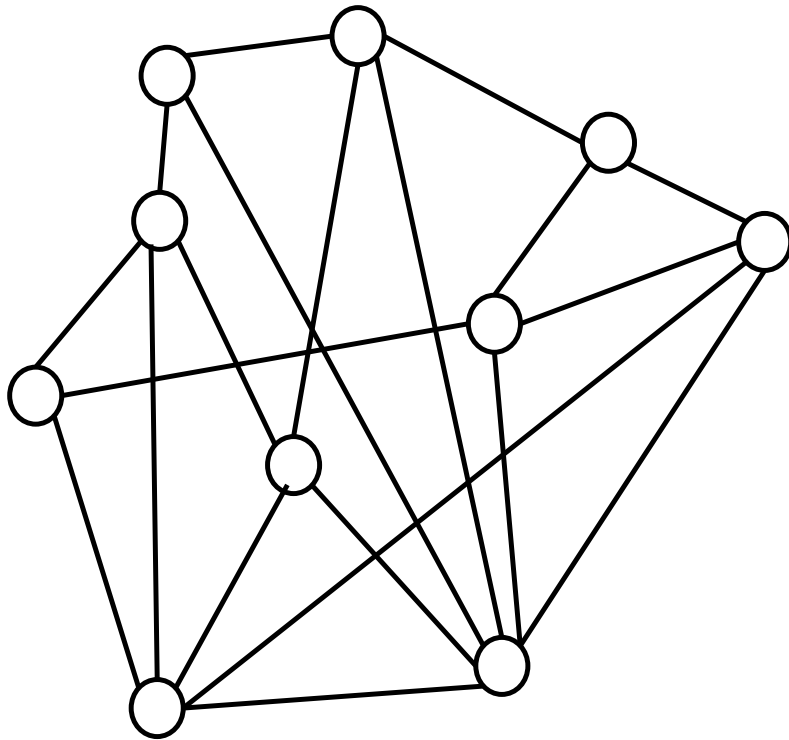
# Ausnahmen

- Entwicklerversionen
- Prototypen

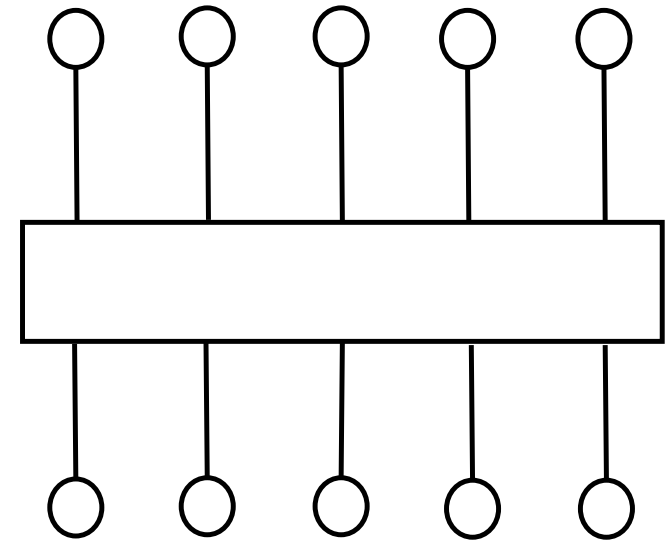
# Lösung

- Komponentenmodell
- mgl. weniger Interfaces als Implementationen
- Vereinheitlichte Interoperabilität (z.B. CORBA)
- Meta-Daten (Naming & Trading Services)

# Beispiel



Stovepipe System



Lösungsansatz

# Verwandte AntiPatterns

- Vendor Lock-In
- Walking through a Minefield

# Schluß



Vielen Dank für die Aufmerksamkeit!

Fragen?