

Event-Driven Objects in .NET

Framework zur Verteilung und
Synchronisation von Objekten

Inhalt

- Ziele
- Problemumfeld
- Begriffe
- Szenario
- Architektur
- Referenzanwendung
- Ergebnisse
- Zusammenfassung und Ausblick

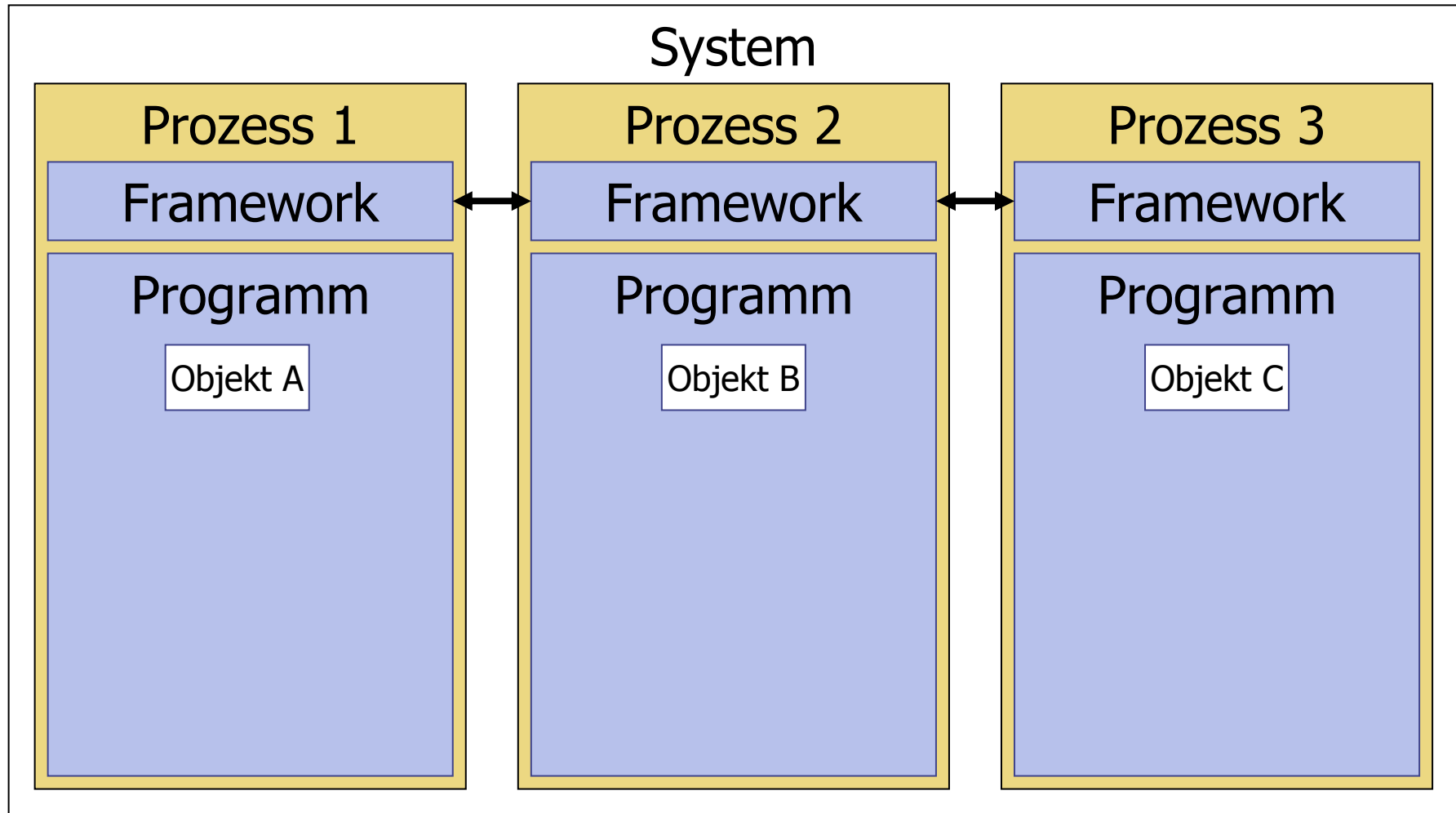
Ziele

- Verteilung und Synchronisation von Objekten
- Dezentrale Organisation der Knoten
- Transaktionssichere Änderungen
- Einfache Einbindung in bestehende Applikationen
- .NET-Plattform

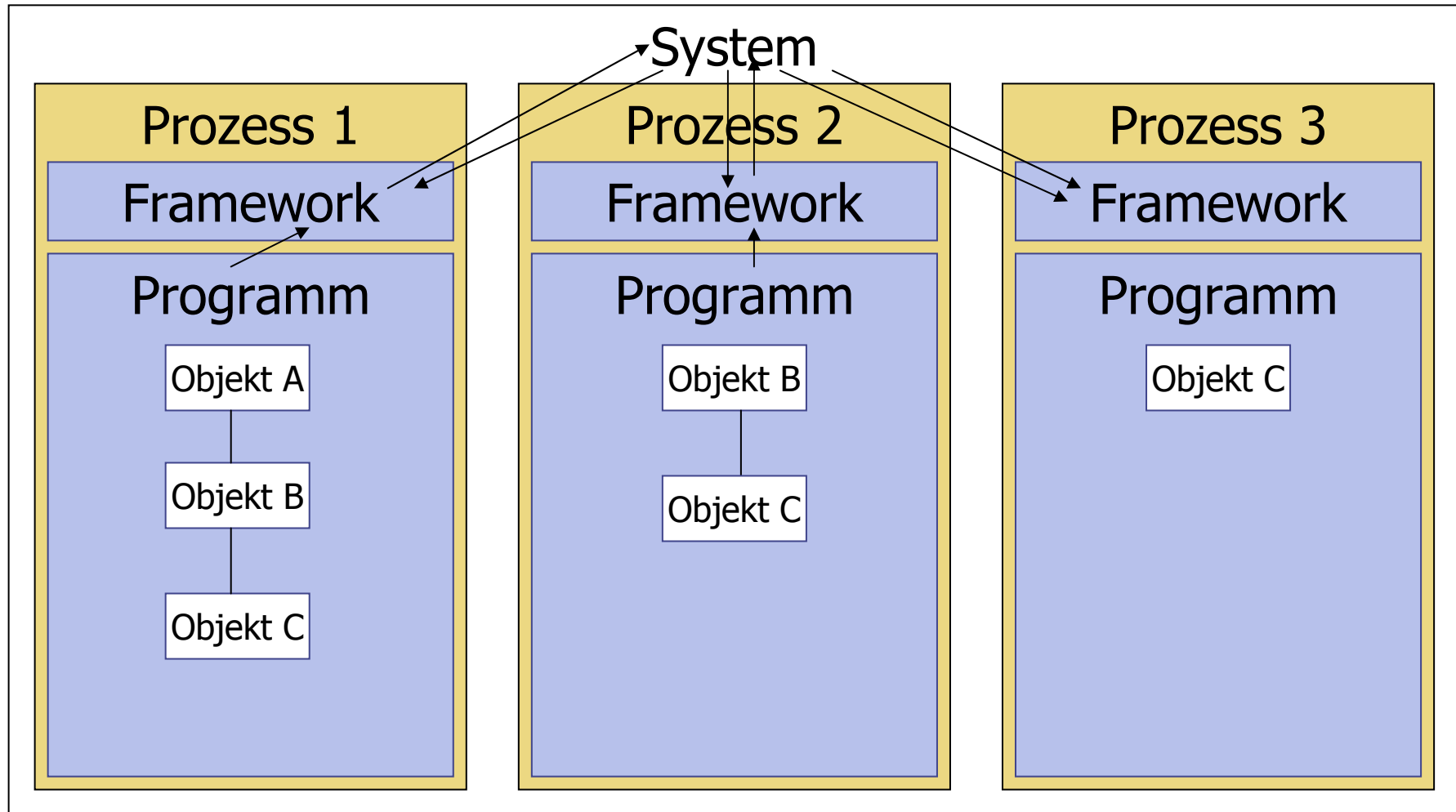
Problemumfeld

- Peer-to-Peer Systeme
- Windows Communication Foundation
- Transaktions Management
- Aspektorientierte Programmierung
- Ereignisgesteuerte Programmierung

Begriffe



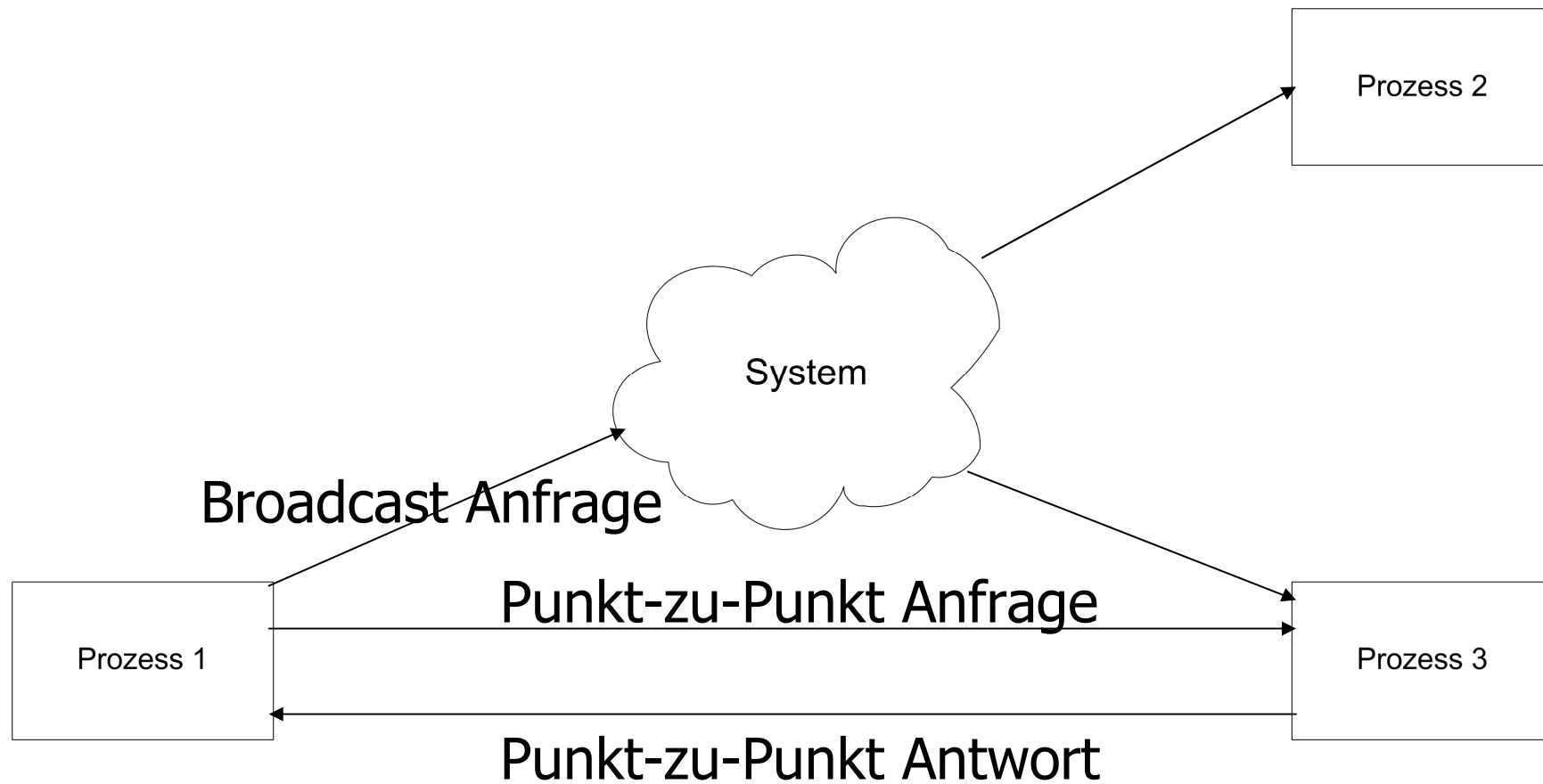
Szenario



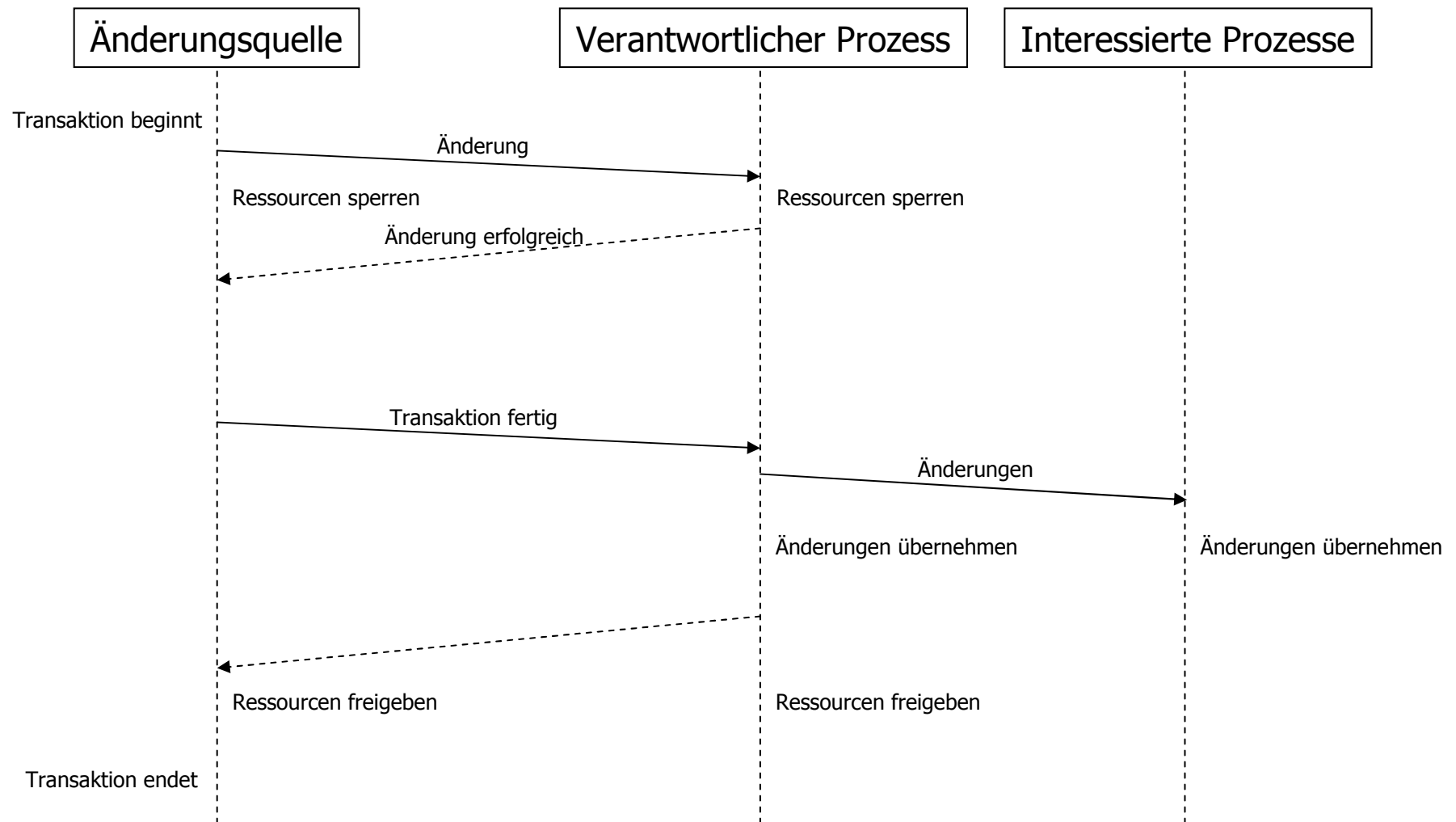
Architektur - Verteilte Objekte

- Eindeutigkeit (ObjectID)
- Generische Manipulation
- Einweben von Events
- „Lazy Loading“

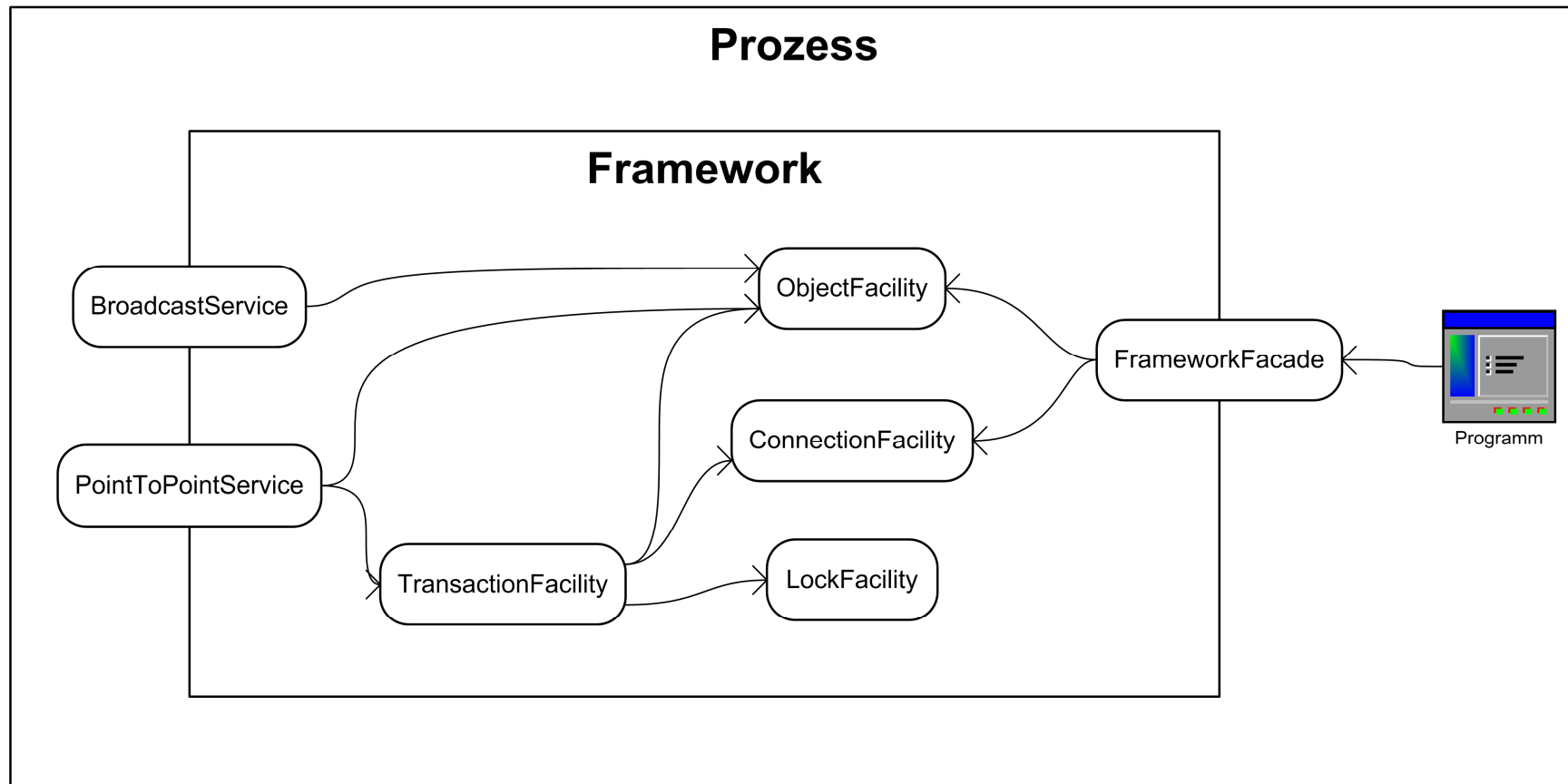
Architektur – Kommunikation



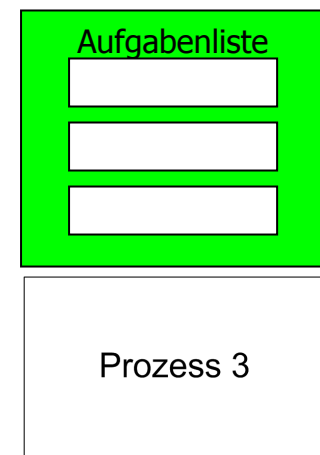
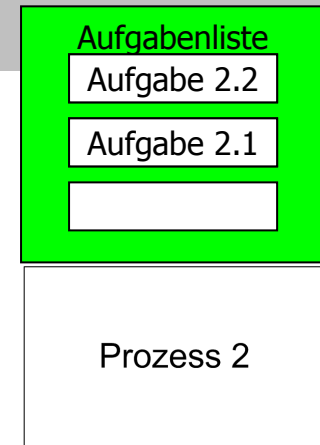
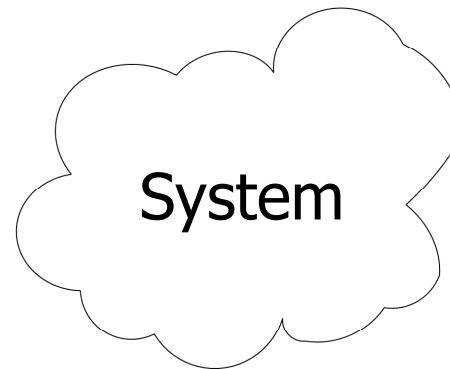
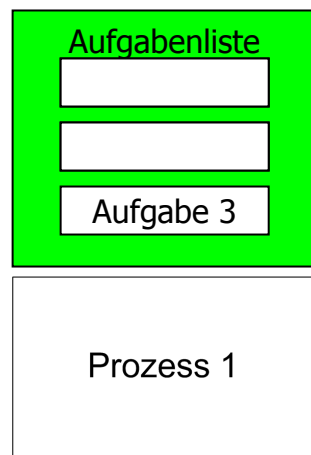
Architektur – Änderungen



Architektur – Komponenten



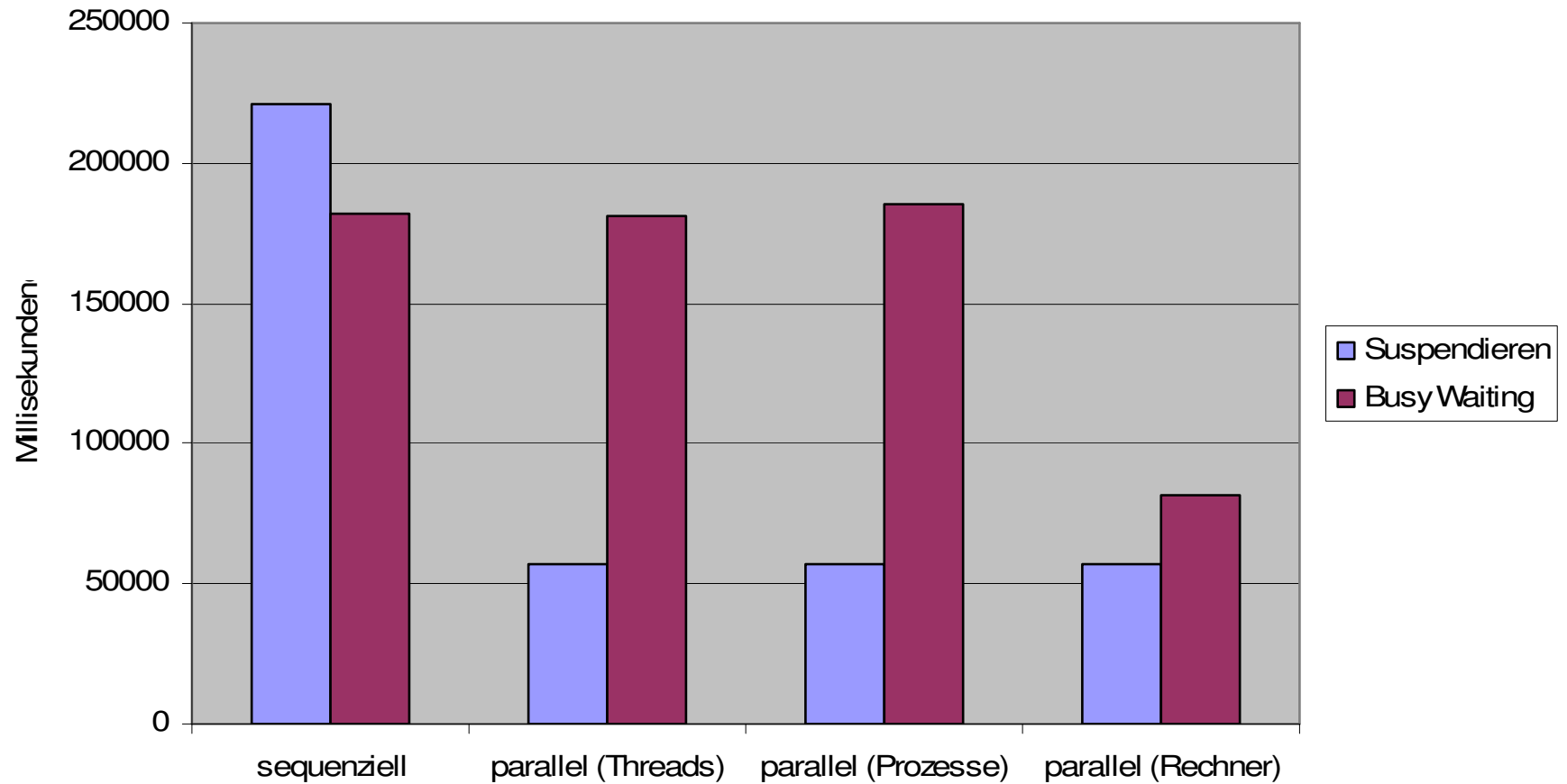
Referenzanwendung



Referenzanwendung – Testszenarien

- Sequenzielle Abarbeitung
- Parallele Abarbeitung 1 Prozess,
4 Threads
- Parallele Abarbeitung 4 Prozesse,
1 Rechner
- Parallele Abarbeitung 4 Prozesse,
4 Rechner

Ergebnisse – Laufzeit



Ergebnisse – Kommunikationsanteil

	Suspendieren	Busy Waiting
sequenziell	0,09%	0,04%
parallel (Threads)	0,30%	0,41%
parallel (Prozesse)	3,40%	2,82%
parallel (Rechner)	3,01%	2,38%

Zusammenfassung und Ausblick

- Framework zur Verteilung und Synchronisation von Objekten
- Referenzanwendung

- Entfernen von Prozessen aus dem System
- Speicherfreigabe
- Finden von Objekten