

# Smart Cluster Ansatz: Open Connectivity & dezentrale Intelligenz

## Was ist ein Smart Cluster?

### Smart Cluster = Open Connectivity + Connected Intelligence

Um wirklich offene, flexible und zukunftssichere Informationssysteme zu bauen, braucht es zwei Dinge: Additivität und dezentrale organisierte Intelligenz.

Alles dreht sich um eine einfache Idee: eine flexible IT-Infrastruktur unterstützt durch dezentrale Intelligenz.

## Die Herausforderung heutiger Ansätze

Ist Ihr Umfeld geprägt von herstellereinspezifischen Insel-Lösungen?  
Möchten Sie starre Strukturen und Systeme öffnen und damit neue Synergien schaffen?  
Sollen auch Ihre Netzwerk Teilnehmer zu einem intelligenten Systemverbund formiert werden?  
Wollen auch Sie eine IT, die zukunftsopen, intelligent und sicher ist?

Derzeit sind IoT - Systeme – und ganz besonders der -Markt – geprägt von einer Vielzahl meist herstellereinspezifischer Systeme und Geräte.

Eine weitere Herausforderung zahlreicher IT-Landschaften: Es handelt sich oft um bis ins Monströse gewachsene, meist geschlossene Systeme. Diese lassen sich meist nur mit großem Aufwand und dem nicht unerheblichen Risiko auf neue Gegebenheiten anpassen.

## Die Lösung steckt in unserem besonderen Ansatz

Wir verfolgen den adaptiven Ansatz: Wir setzen auf herstellerübergreifende und somit offene Standards. Mit smarten kleinen Komponenten (wie z.B. Microservices) und transparenten Schnittstellen (Event-Bus, RESTful APIs) können Sie endlich wieder preisgünstig und flexibel auf geänderte Anforderungen reagieren.

Wir arbeiten ganzheitlich: Berücksichtigen alle beteiligten Schichten mit ihren diversen Standards und setzen auf Automatisierung.

Die Stärke unseres Ansatzes liegt in der Verwendung einer gemeinsamen Konfiguration, dem "Infrastruktur Repository". Hier sind alle beteiligten Systeme und Komponenten hinterlegt. Bei Veränderungen werden alle betroffenen Teile vollautomatisch über das neue Setting informiert. So kann stets gewährleistet werden, dass alle Teilnehmern sicher, fehlertolerant und ganzheitlich kommunizieren können, bei voller Transparenz und vollständiger Kontrolle.

So kann Ihre Technik , zeitnah und flexibel auf Veränderungen reagieren, ohne jeweils aufwendig und damit teuer angepasst zu werden.

## Was man durch ein Smart Cluster gewinnt

Um wirklich offene, flexible und zukunftssichere Informationssysteme zu bauen, gilt unbedingt es die folgenden Grundprinzipen zu berücksichtigen:

- **Technologieneutral**  
Gestalte alle Schnittstellen Technologieneutral (somit unabhängig von Plattform und Programmiersprache)
- **Einheitliche Schnittstellen**  
Verwende stets allgemein gültige Funktionen und strukturierte Daten (Document Style). Hier lohnt ein Blick auf das RESTful Design Prinzip.
- **Stateless**  
Vermeide Sessions, Cookies, Tracking Parameter und andere implizite Status-Speicher. Denn sie blockieren Ressourcen, verschleiern die Kommunikationsflüsse, erschweren die Nutzung und gefährden die Sicherheit.
- **Transparenz**  
Sei offen. Stelle die API jedem zur Verfügung. Verwende eine gemeinsame Infrastruktur Datenbank, aus der alle Konfigurationen entnommen und in die alle aktiv genutzten Schnittstellen mit ihren Parametern geschrieben werden.
- **Schichtentrennung**  
Für jede Schicht die passende Technologie wählen. Etablierte Standards nutzen, wo immer möglich. Nicht das Rad neu erfinden.
- **Data Centric**  
Wo immer möglich ausschließlich fachlich orientierte Daten(strukturen) verwenden. Dann wird die API die natürliche Basis für kommende Fachgespräche.

## Skalierbar

Skalierbarkeit heißt Parallelität. Am besten in kleinen Einheiten (wie z. B. Microservices). Das ist das Gegenteil von monolithischen Systemen.

## Event-Bus / Nachrichten-Bus / Message-Bus

Für die zuverlässige Kommunikation in vernetzten Systemen ist ein (publish-subscribe) Nachrichten-Bus (wie z. B. MQTT) optimal. Mit einem solchen, können alle Daten und Events in einer Queue für alle interessierten Nodes vorgehalten werden.

# Basiskomponenten der Smart Cluster Strategie

xxx

## Dezentrale Steuerung

Statt alle Ebenen der Steuerung in einer zentralen Plattform zu implementieren, werden von der Steuerlogik nur noch die „Arbeitsaufträge“ an den Adapter des Gerätes / der Maschine geschickt. Dort kann dann das ganze Maschinenspezifische Detailwissen lokal in Steuerbefehle und Messgrößen umgesetzt werden. Insbesondere vereinfacht dies den Aufbau und den Test eines digitalen Zwilling massiv und reduziert maßgeblich die Komplexität. Da die Steuerlogik / Prozesse dann nur noch die logische Abarbeitung implementieren muss und die physische Umsetzung dem Gerätespezifischen Adapter (mit seiner Hardwarespezifischen Software-Version) überlassen werden kann.

## Dezentrale Steuerung

Statt alle Ebenen der Steuerung in einer zentralen Plattform zu implementieren, werden von der Steuerlogik nur noch die „Arbeitsaufträge“ an den Adapter des Gerätes / der Maschine geschickt. Dort kann dann das ganze Maschinenspezifische Detailwissen lokal in Steuerbefehle und Messgrößen umgesetzt werden. Insbesondere vereinfacht dies den Aufbau und den Test eines digitalen Zwilling massiv und reduziert maßgeblich die Komplexität. Da die Steuerlogik / Prozesse dann nur noch die logische Abarbeitung implementieren muss und die physische Umsetzung dem Gerätespezifischen Adapter (mit seiner Hardwarespezifischen Software-Version) überlassen werden kann.

## Infrastruktur Repository

Das Infrastruktur Repository dient zur zentralen Konfiguration aller Gerätespezifischen Parameter. In diesem werden auch alle Kommunikationspfade mit deren Security Werten erfasst. Dies ist die Basis für ein (regelmäßiges) auto-Deployment aller beteiligten Komponenten.

Auch ist es die basis für ein automatisiertes Intrusion Detection und zur Falsifizierung von (Daten-)Bewegungsmustern.


So ist es leicht möglich, unterschiedliche Software-Umgebungen vollständig und sauber voneinander separiert zu betreiben. Hierzu gehören z. B. (Dev, Test und Prod-Umgebung), digitaler Zwilling oder das Disaster Recovery.

Als Nebeneffekt erhält man so die maximale Transparenz aller Komponenten mit deren Kommunikationsverbindungen, Software Versionen, Security Credentials und Recovery Daten.




### Im Ergebniss ergeben sich die folgenden Freiheiten:

- Heteroge ITK und IT Service Landschaft --> den maximalen Freiheitsgrad für künftige Anpassungen
- Sanfte Migration (statt big-bang Deployment oder single-version PROD)
- Dezentrale Steuerung --> Maschienspezifische Kenntnisse können im Adapter des Gerätes / Maschine von den Geräte-Experten umgesetzt werden. Statt dieses Know-How erst umständlich auf eine zentrale Plattform zu migrieren.
- Homogenes Verhalten aller beteiligten Systeme
- Maximale Trasparenz der IT/ITK Lanschaft und seiner Verknüpfungen




**Unternehmen**

 bintellix GmbH & Co. KG  
# Geigenbergerstr. 7a  
# 81477 München  
# Deutschland

**Comunity**

 facebook.com/bintellix  
 twitter.com/bintellix  
 github.com/twitter

**Kontakt**

 +49 89-7507504-0  
 +49 89-7507504-99  
 info@bintellix.com  
 Kontaktformular

**Unternehmensgruppe**