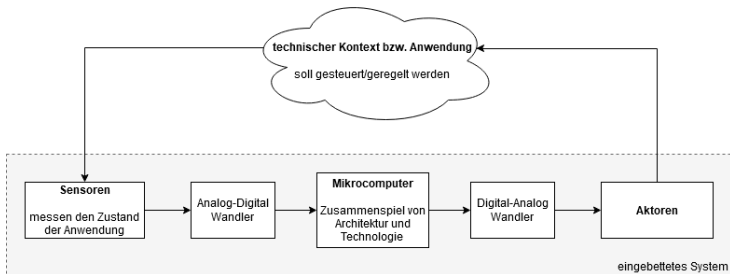




## Generische cross-domain Architektur für eingebettete Systeme

# Was ist ein eingebettetes System?

## Was ist ein eingebettetes System?



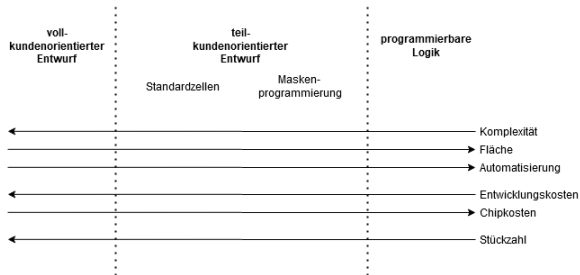
## Wieso generische cross-domain Architektur?

## Wieso generische cross-domain Architektur?

- ▶ Interoperabilität

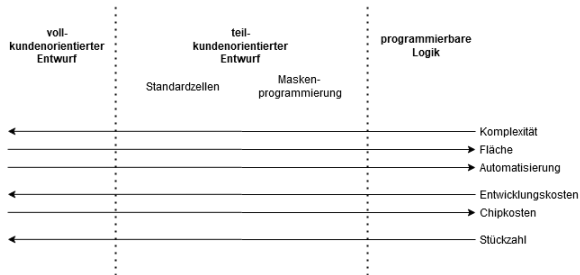
## Wieso generische cross-domain Architektur?

- ▶ Interoperabilität
- ▶ Technologie-Einschränkungen



## Wieso generische cross-domain Architektur?

- ▶ Interoperabilität
- ▶ Technologie-Einschränkungen



- ▶ Personelle Ressourcen

# Was ist GENESYS?



## Was ist GENESYS?

europäischer Sonderforschungsbereich 501

- ▶ „Entwicklung großer Systeme mit generischen Methoden“

## Was ist GENESYS?

europäischer Sonderforschungsbereich 501

- ▶ „Entwicklung großer Systeme mit generischen Methoden“

Teilprojekt B5

- ▶ „GENerische SYStemsoftware“
- ▶ generische Ansätze für eingebettete Systeme
  - ▶ Generische Softwarekomponenten als Basis
  - ▶ Einsatz von Generatortechniken

## Generizität

Konzept, bei dem eine Struktur in variante und invariante Teile unterteilt wird.

## Generizität

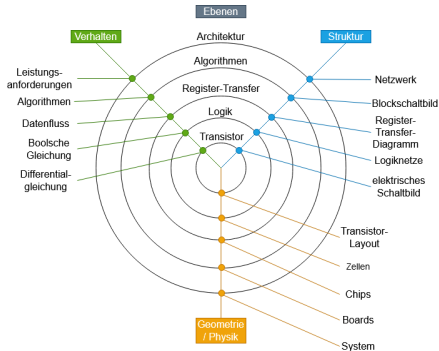
Konzept, bei dem eine Struktur in variante und invariante Teile unterteilt wird.

Erstellung einer anwednungsspezifischen Laufzeitplattform:

- ▶ Auswahl
- ▶ Konfiguration
- ▶ Kombination

## Generizität am Beispiel von VHDL

```
generic (name_of_generic_variable :  
          type_of_generic_variable :=  
          default_value_of_generic_variable);
```



## ARTEMIS Strategic Research Agenda (2006)

Herausforderungen für generische cross-domain  
Architektur:

## ARTEMIS Strategic Research Agenda (2006)

Herausforderungen für generische cross-domain Architektur:

1. Kombinierbarkeit

## ARTEMIS Strategic Research Agenda (2006)

Herausforderungen für generische cross-domain Architektur:

1. Kombinierbarkeit
2. Vernetzung



## ARTEMIS Strategic Research Agenda (2006)

Herausforderungen für generische cross-domain Architektur:

1. Kombinierbarkeit
2. Vernetzung
3. Sicherheit

## ARTEMIS Strategic Research Agenda (2006)

Herausforderungen für generische cross-domain Architektur:

1. Kombinierbarkeit
2. Vernetzung
3. Sicherheit
4. Robustheit

## ARTEMIS Strategic Research Agenda (2006)

Herausforderungen für generische cross-domain Architektur:

1. Kombinierbarkeit
2. Vernetzung
3. Sicherheit
4. Robustheit
5. Diagnose und Wartung

## ARTEMIS Strategic Research Agenda (2006)

Herausforderungen für generische cross-domain Architektur:

1. Kombinierbarkeit
2. Vernetzung
3. Sicherheit
4. Robustheit
5. Diagnose und Wartung
6. Integriertes Ressourcen-Management

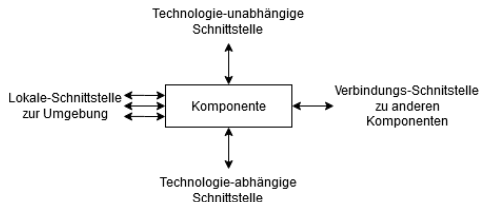
## ARTEMIS Strategic Research Agenda (2006)

Herausforderungen für generische cross-domain Architektur:

1. Kombinierbarkeit
2. Vernetzung
3. Sicherheit
4. Robustheit
5. Diagnose und Wartung
6. Integriertes Ressourcen-Management
7. Entwicklungsfähigkeit

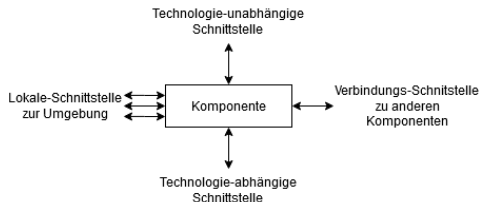
## GENESYS: Architekturprinzipien I

### ► Strikte Komponenten-Orientierung



## GENESYS: Architekturprinzipien I

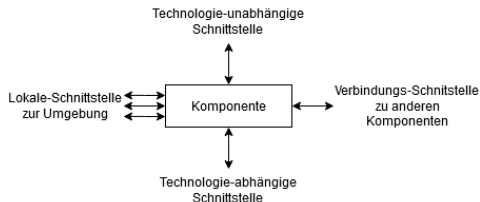
### ► Strikte Komponenten-Orientierung



### ► Strikte Trennung von Kommunikation und Berechnung

## GENESYS: Architekturprinzipien I

### ► Strikte Komponenten-Orientierung

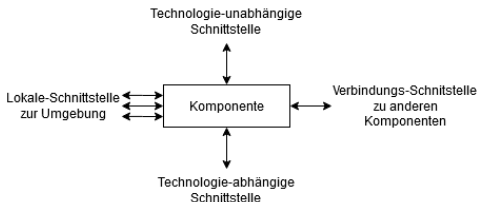


- Strikte Trennung von Kommunikation und Berechnung
- Kommunikation

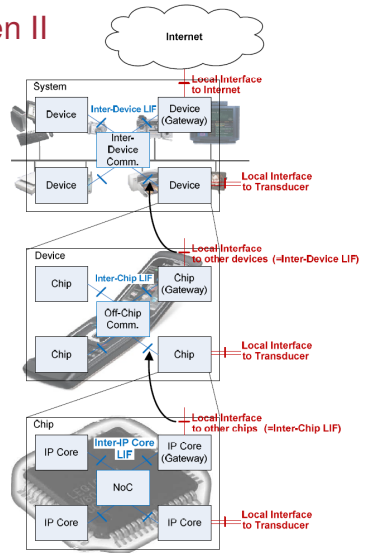


## GENESYS: Architekturprinzipien I

### ► Strikte Komponenten-Orientierung

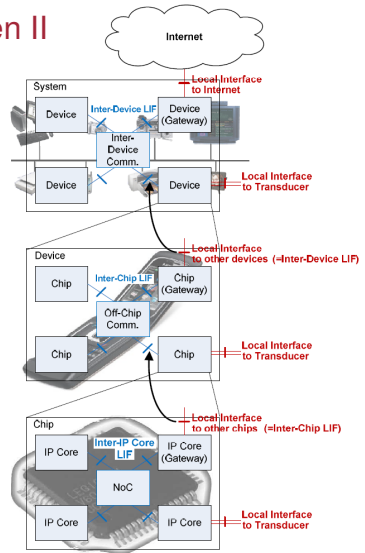


- Strikte Trennung von Kommunikation und Berechnung
- Kommunikation
- Verfügbarkeit einer gemeinsamen Zeit



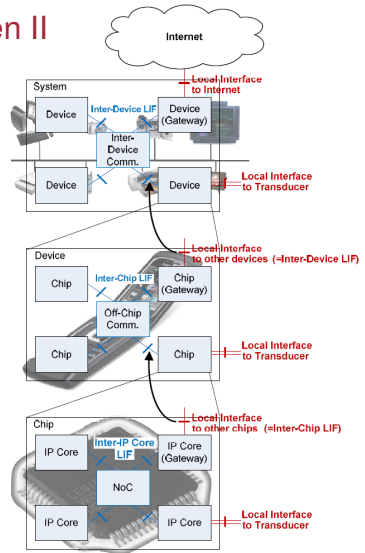
## GENESYS: Architekturprinzipien II

- Hierarchische Systemstruktur
- Zustandsbewusstsein



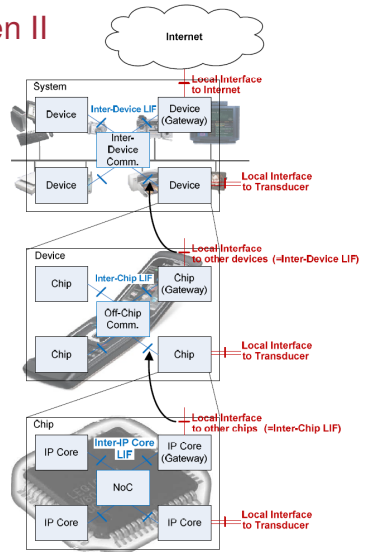
## GENESYS: Architekturprinzipien II

- Hierarchische Systemstruktur
- Zustandsbewusstsein
- Fehlerisolation



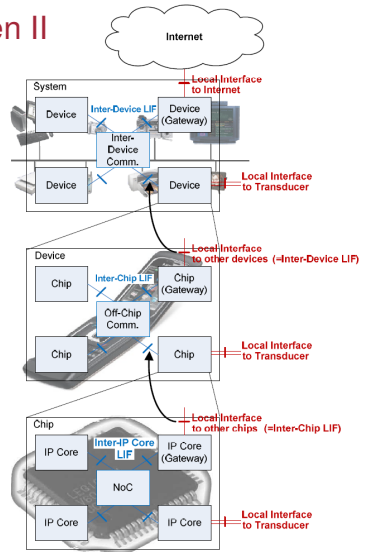
## GENESYS: Architekturprinzipien II

- Hierarchische Systemstruktur
- Zustandsbewusstsein
- Fehlerisolation
- Wissen und Kontrolle über Ressourcen

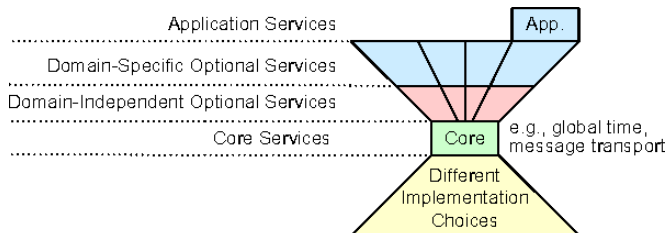


## GENESYS: Architekturprinzipien II

- Hierarchische Systemstruktur
- Zustandsbewusstsein
- Fehlerisolation
- Wissen und Kontrolle über Ressourcen
- Struktur von Diensten



## GENESYS: Architektur



Kern-Services: Zeit-Dienst, Kommunikations-Dienst, Ausführungskontroll-Dienst, Konfigurations-Dienst

Domänen-unabhängige optionale Services: Diagnose-Dienst, Sicherheits-Dienst, ...

Domänen-spezifische optionale Services: CAN Kommunikation, ...

## Erfüllung der ARTEMIS Herausforderungen

	<b>ARTEMIS Herausforderungen</b>						
<b>GENESYS Architekturprinzipien</b>	1	2	3	4	5	6	7
Strikte Komponenten-Orientierung	++	+	+	++	++	++	++
Strikte Trennung von Kommunikation und Berechnung	++	++	+	++	++	+	++
Verfügbarkeit einer gemeinsamen Zeit	++	+	+	+	+		
Hierarchische Systemstruktur	++		+	+	+	+	+
Kommunikation	++	++	+	+	+	+	+
Zustandsbewusstsein	+		+	++	++	+	+
Fehlerisolation	++		++	++	++		+
Wissen und Kontrolle über Ressourcen	+	+		+	+	++	+
Struktur von Services	++	++	+	+	+		++

ARTEMIS Herausforderungen: 1 Kombinierbarkeit

2 Vernetzung

3 Sicherheit

4 Robustheit

5 Diagnose und Wartung

6 Integriertes Ressourcen-Management

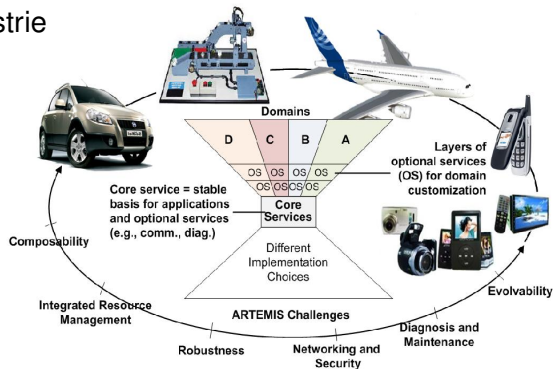
7 Entwicklungsfähigkeit



## Praktische Anwendung: INDEXYS

### INDustrielle EXplorierung von geneSYS

- ▶ Automobil-Industrie
- ▶ Flugzeug-Industrie
- ▶ Bahn-Industrie
- ▶ ...



## Literatur I



Oliver Bringmann, Walter Lange und Martin Bogdan. *Eingebette Systeme - Entwurf, Modellierung und Synthese*. Berlin: Walter de Gruyter GmbH & Co KG, 2018. ISBN: 978-3-110-51852-8.



Sorin Cotofana, Stephan Wong und Stamatios Vassiliadis. *Embedded Processors: Characteristics and Trends*. Techn. Ber. Delft University of Technology, März 2004.



Andreas Eckel u. a. „INDEXYS, a Logical Step beyond GENESYS“. In: *Proceedings of the 29th International Conference on Computer Safety, Reliability, and Security*. Aug. 2010, S. 431–451.



Christoph M. Kirsch und Raja Sengupta. „The Evolution of Real-Time Programming“. In: *Handbook of Real-Time and Embedded Systems*. 2007.



Andreas Mäder. *VHDL Kompakt*. Universität Hamburg: Fakultät für Mathematik, Informatik und Naturwissenschaften.



Jürgen Münch und Bernd Schürmann. „Sonderforschungsbereich 501: Entwicklung großer Systeme mit generischen Methoden“. In: *it - Information Technology* 45 (Apr. 2003), S. 227–236.

## Literatur II



Roman Obermaisser und Hermann Kopetz. „From ARTEMIS Requirements to a Cross-Domain Embedded System Architecture“. In: *ERTS2 2010, Embedded Real Time Software & Systems*. Mai 2010.



Roman Obermaisser u. a. „Fundamental Design Principles for Embedded Systems: The Architectural Style of the Cross-Domain Architecture GENESYS“. In: *2009 IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*. März 2009, S. 3–11.



Prof. Dr.-Ing. Frank Slomka. *Grundlagen der Rechnerarchitektur - Von der Schaltung zum Prozessor*. Universität Ulm: Institut für eingebettete Systeme/Echtzeitsysteme, 2020.



Prof. Dr.-Ing. Frank Slomka. *Vorlesung Architektur eingebetteter Systeme*. Sommersemester 2020.

## Übung zur Vorlesung

+

praktische Arbeit

Zuweisungen finden erst  
am Ende des process statt

alternativen finden: getchar  
statt scanf verwenden, ...

-

Tool- / Software-bedingte  
Fehler

← process läuft sequentiell ab

← nicht alle C-Funktionen wie  
gewohnt verwendbar

⇒ viel gelernt: VHDL

Problemlösung

eigenständiges Arbeiten