


# Embedded Software Manager Pattern

## Content

	MANAGER
Content .....	1
1 Content.....	3
2 Software System Wide Tasks .....	4
2.1 Manager Concept .....	5
2.1.1 Principle .....	5
2.2 Applying to Increasing Software Complexity .....	6
2.2.1 Architecture Levels: Infinite Multiplicity .....	6
2.2.2 Architecture Levels: Infinite Depth .....	7
2.2.3 Architecture Level Tree .....	8
2.3 Initialization .....	9
2.3.1 Procedural Programming Approach .....	9
2.4 Instantiation and Initialization .....	10
2.4.1 Object Oriented Programming Approach .....	10
2.5 Relation Initialization .....	11
2.6 Configuration, Parametrization, Data Base Access .....	12
2.7 Resource Pre-Allocation .....	13
2.8 Boot, Start and Run .....	14
2.8.1 Without Operating System (Bare Metal) .....	14
2.8.2 With Operating System .....	15
2.9 Diagnostics .....	16
2.10 Restart .....	17
2.11 Shutdown .....	18
2.12 Error Management .....	19
2.12.1 Central .....	19

2.12.2	Decentral .....	20
2.13	Manager State Machine .....	21
2.14	Manager Top Level Execution Flow .....	22
2.15	Discussion Topics and Improvements .....	23
2.15.1	Basic Concept with Error Notification Interface .....	24
2.15.2	Extended Concept with Error Notification and Manager Interface .....	25
3	Summary and Outlook.....	26
4	MicroConsult Training and Coaching.....	27

# 1 Content

Content
▪ Software system wide tasks
▪ Manager concept principle
▪ Applying to increasing software complexity
▪ Manager state machine
▪ Manager execution flow
▪ Discussion topics and improvements
▪ Basic and extended concept
▪ Summary
<small>© MicroConsult - Microelectronics Consulting &amp; Training GmbH</small> <small>Tuesday, September 5, 2023</small> <small>S 2</small> 


## 2 Software System Wide Tasks

**Software System Wide Tasks**

The following software system wide **tasks** are relevant for **each architecture element** and need to be coordinated **overall**:

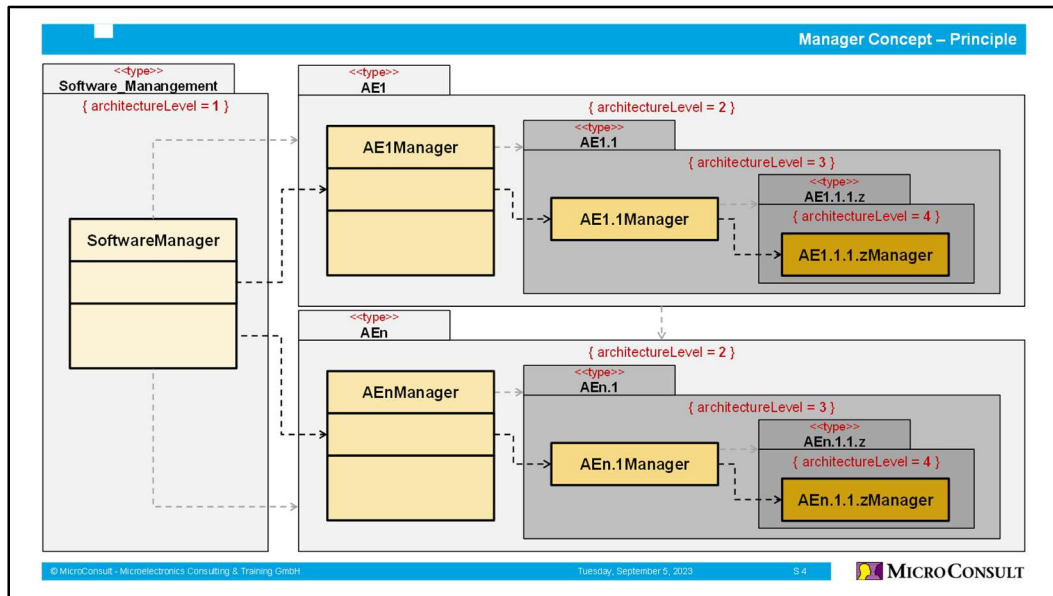
- Software system / subsystem **instantiation and initialization**
- Software system / subsystem **relation initialization**
- Software system / subsystem **configuration, parametrization, database access**
- Software system / subsystem **resource allocation**
- Software system / subsystem **boot**
- Software system / subsystem **execution**
- Software system / subsystem **diagnostics, self test, debugging**
- Software system / subsystem **reboot**
- Software system / subsystem **shutdown**
- Software system / subsystem **error management**
- ... and less or further software system wide tasks

These tasks have to be coordinated by applying a **software system wide management concept**.

© MicroConsult - Microelectronics Consulting & Training GmbHTuesday, September 5, 202353 MICROCONSULT

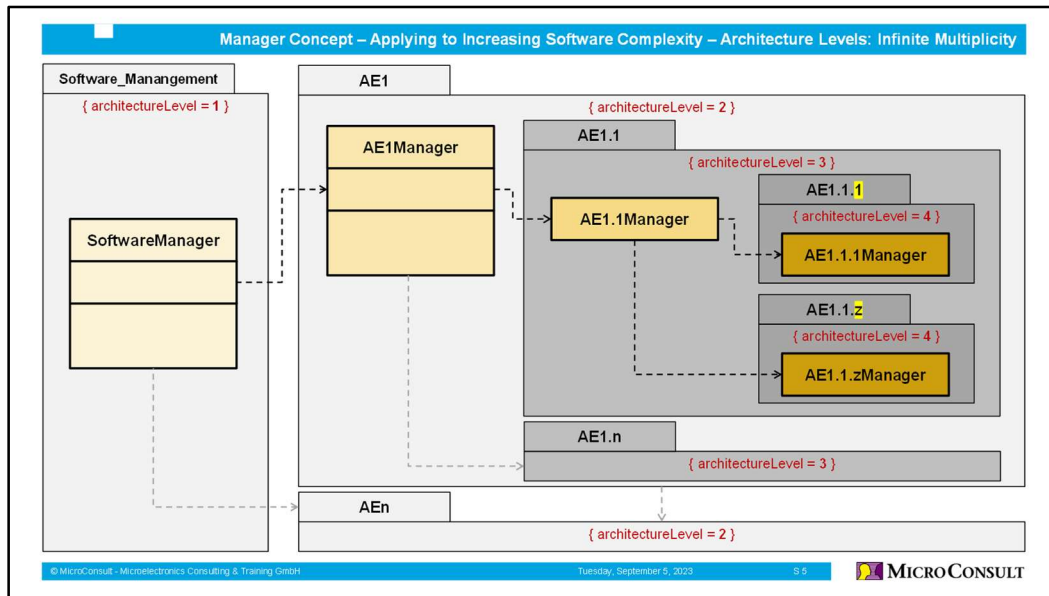
## 2.1 Manager Concept

### 2.1.1 Principle

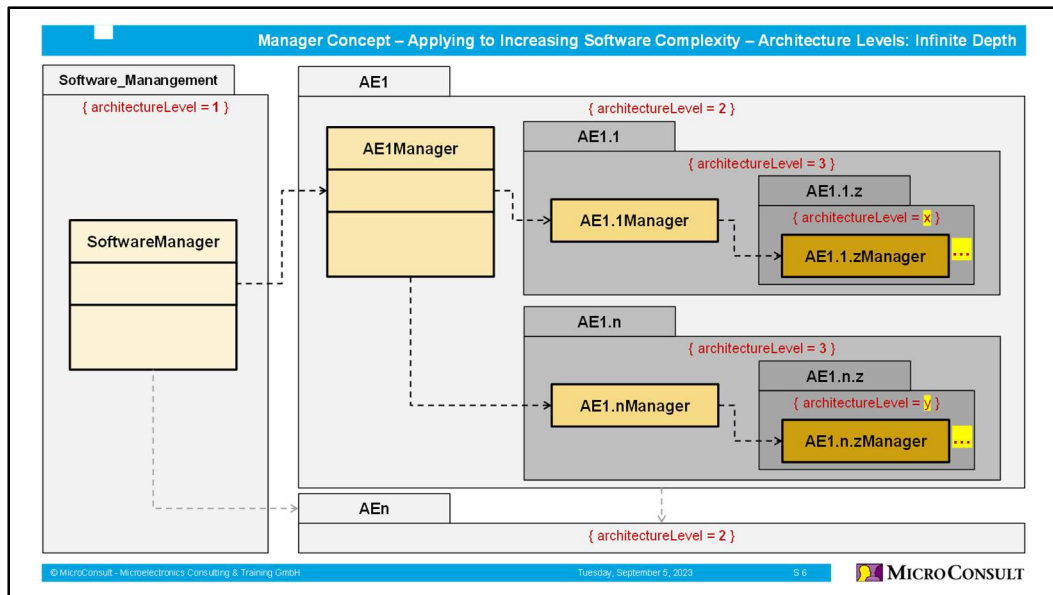


## 2.2 Applying to Increasing Software Complexity

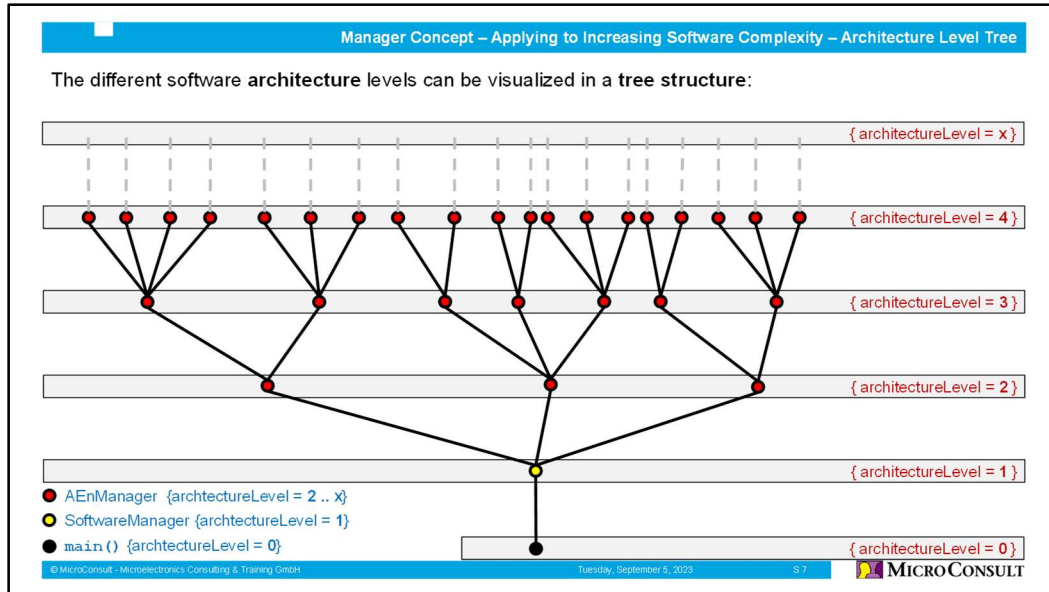
### 2.2.1 Architecture Levels: Infinite Multiplicity



## 2.2.2 Architecture Levels: Infinite Depth



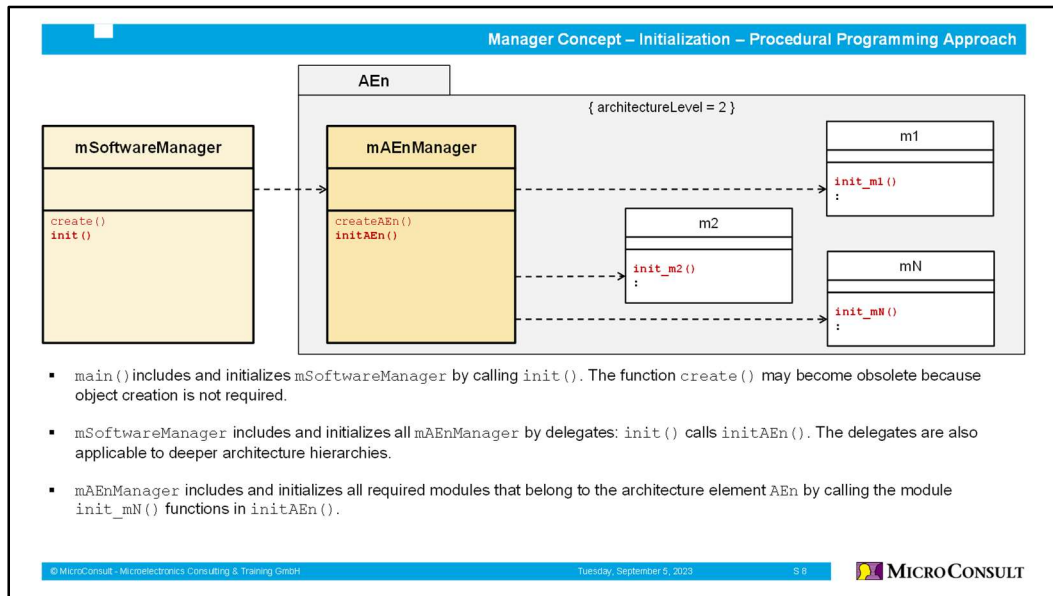
### 2.2.3 Architecture Level Tree





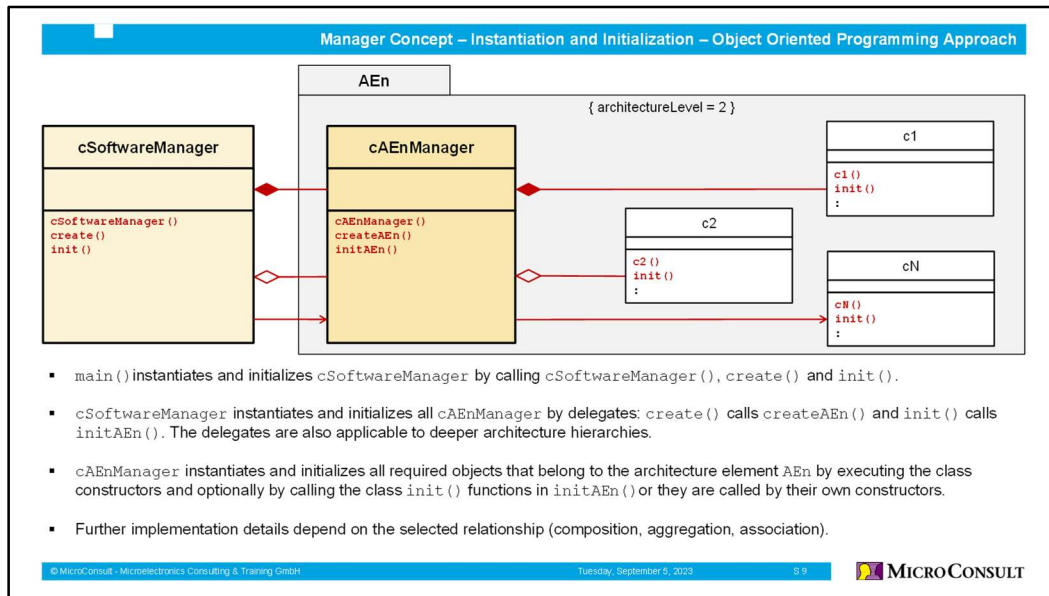
## 2.3 Initialization

### 2.3.1 Procedural Programming Approach

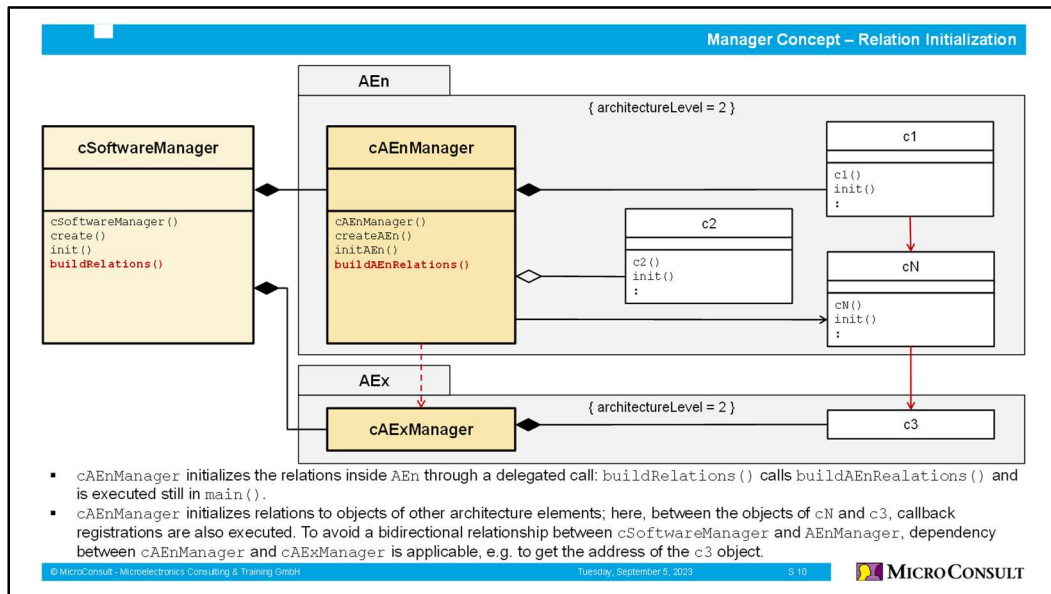


## 2.4 Instantiation and Initialization

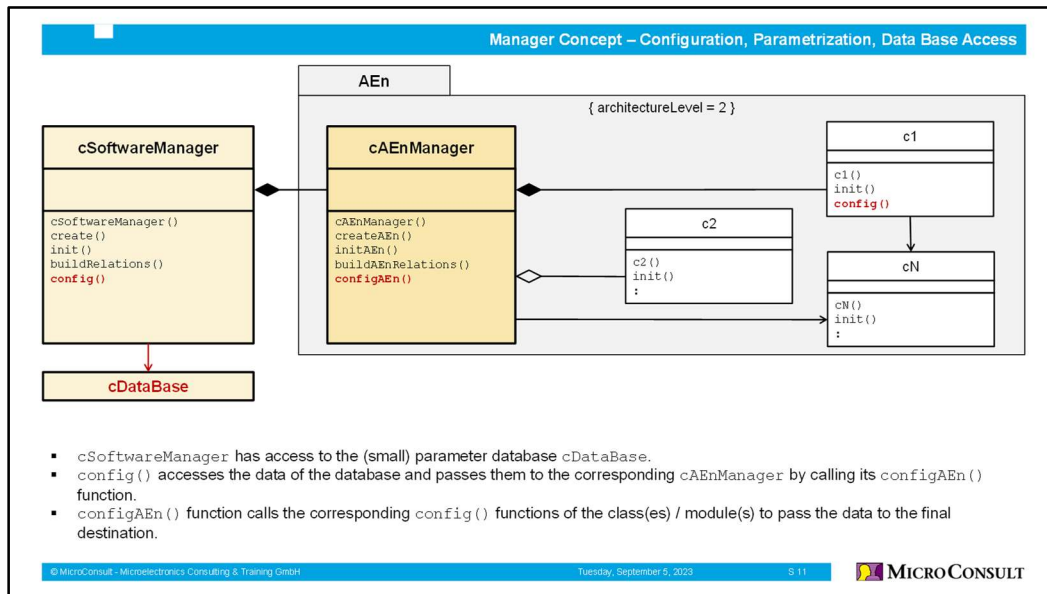
### 2.4.1 Object Oriented Programming Approach



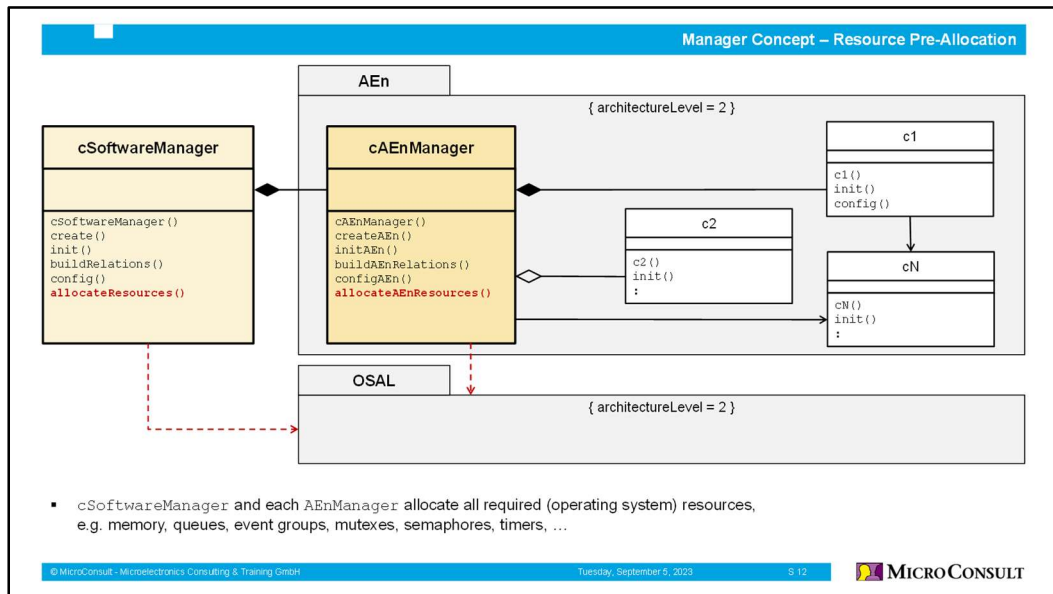
## 2.5 Relation Initialization



## 2.6 Configuration, Parametrization, Data Base Access

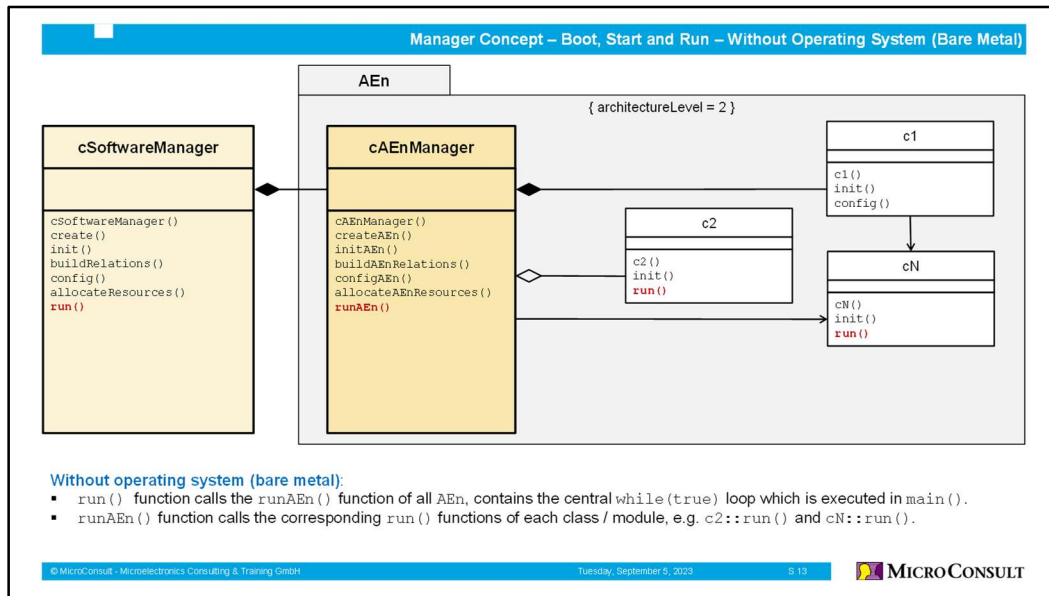


## 2.7 Resource Pre-Allocation

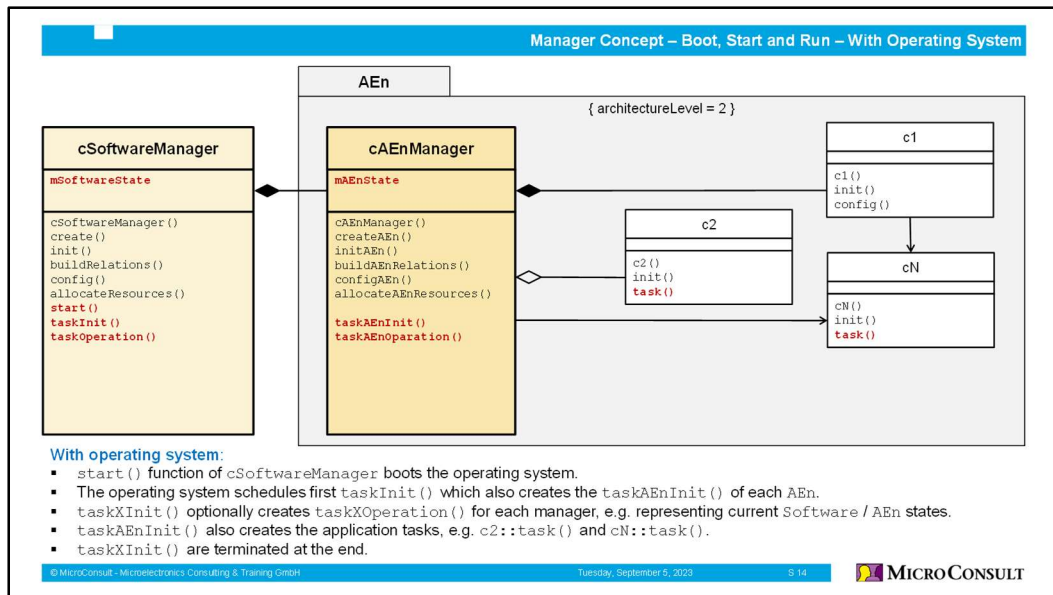


## 2.8 Boot, Start and Run

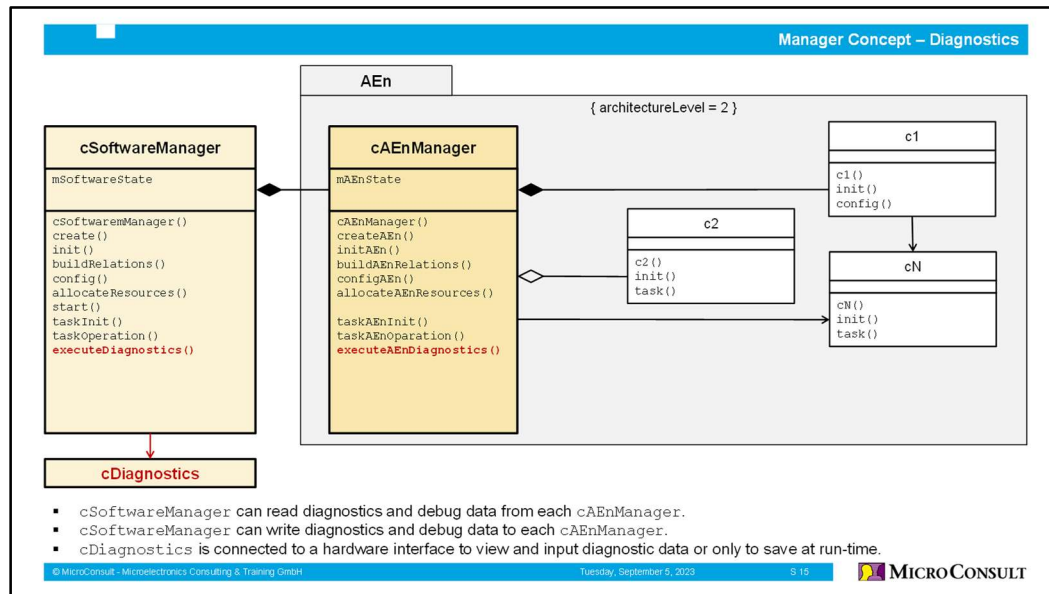
### 2.8.1 Without Operating System (Bare Metal)



## 2.8.2 With Operating System

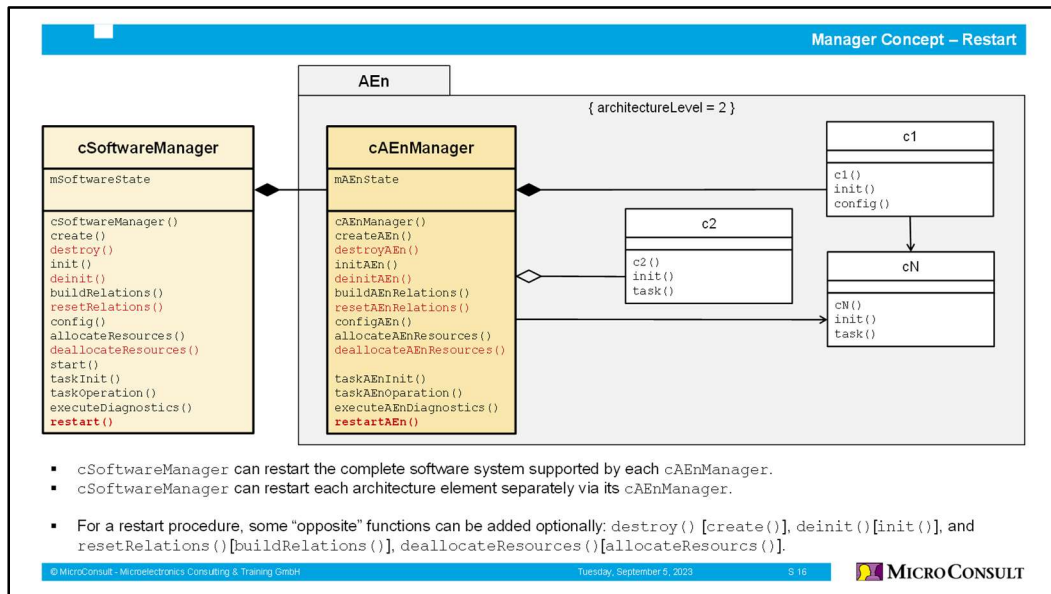


## 2.9 Diagnostics

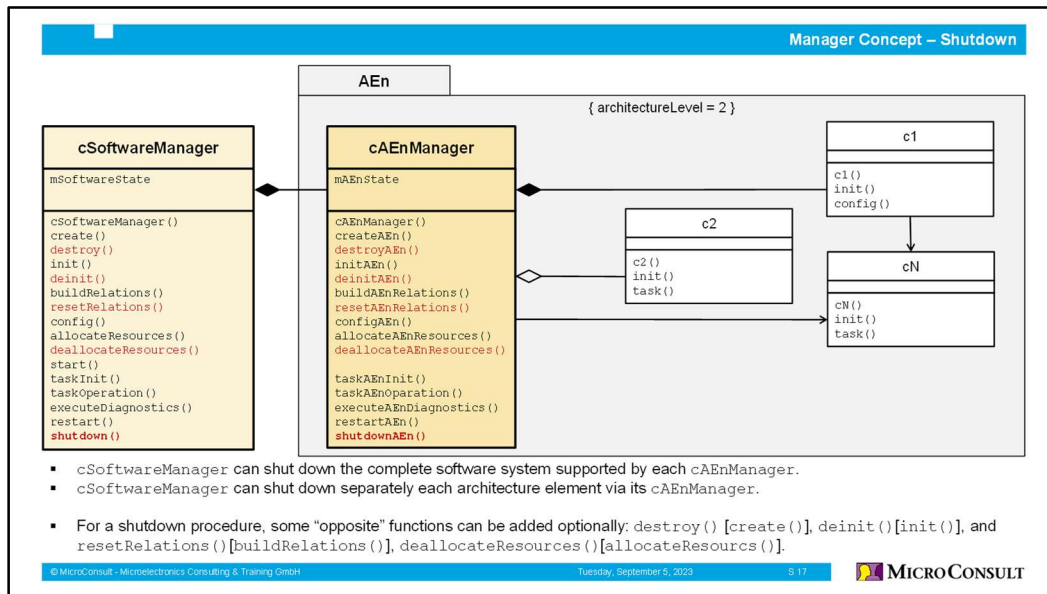




## 2.10 Restart

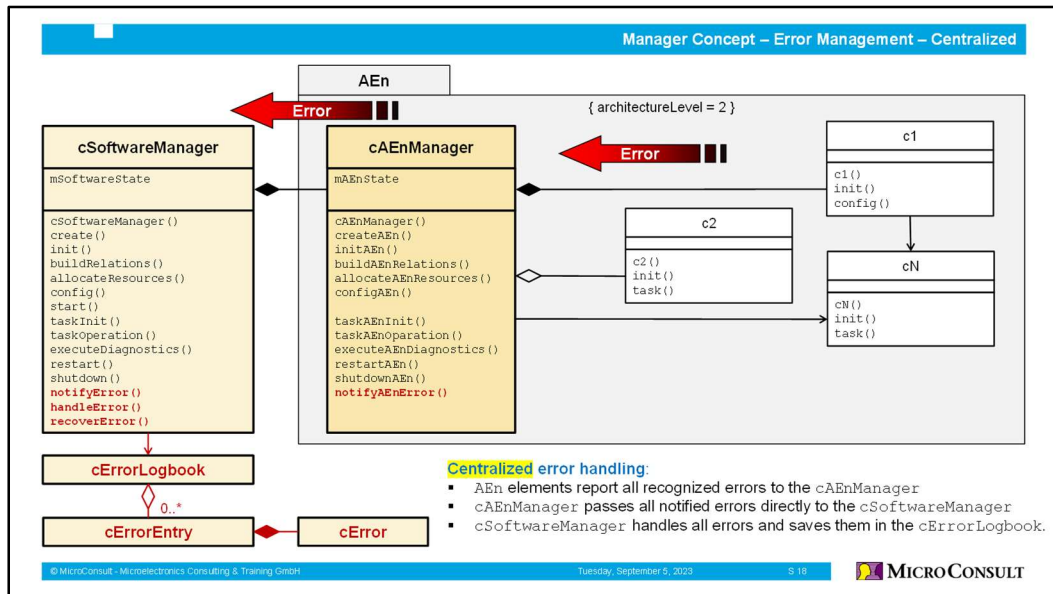


## 2.11 Shutdown

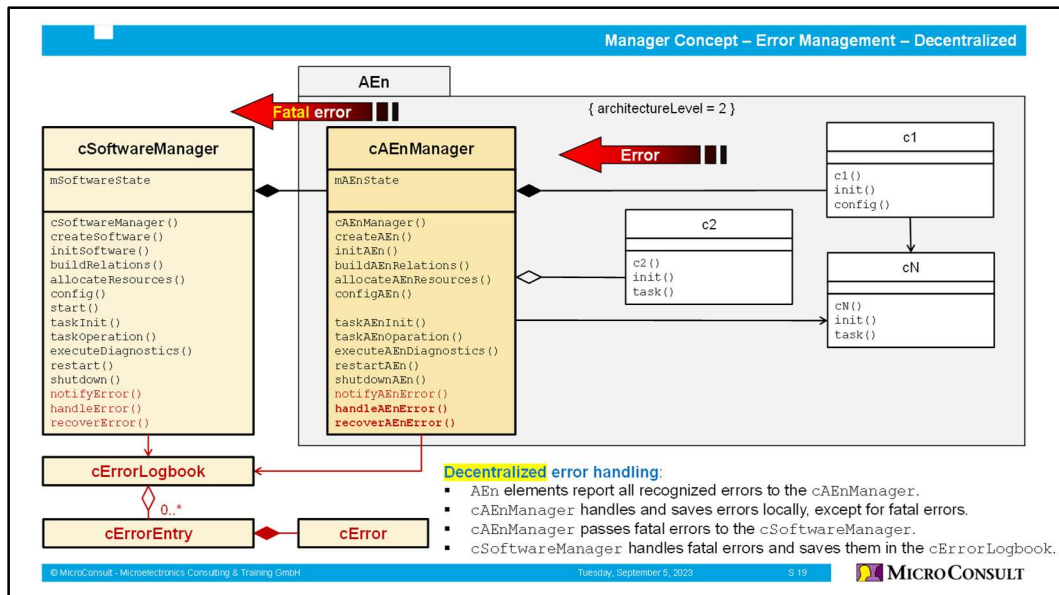


## 2.12 Error Management

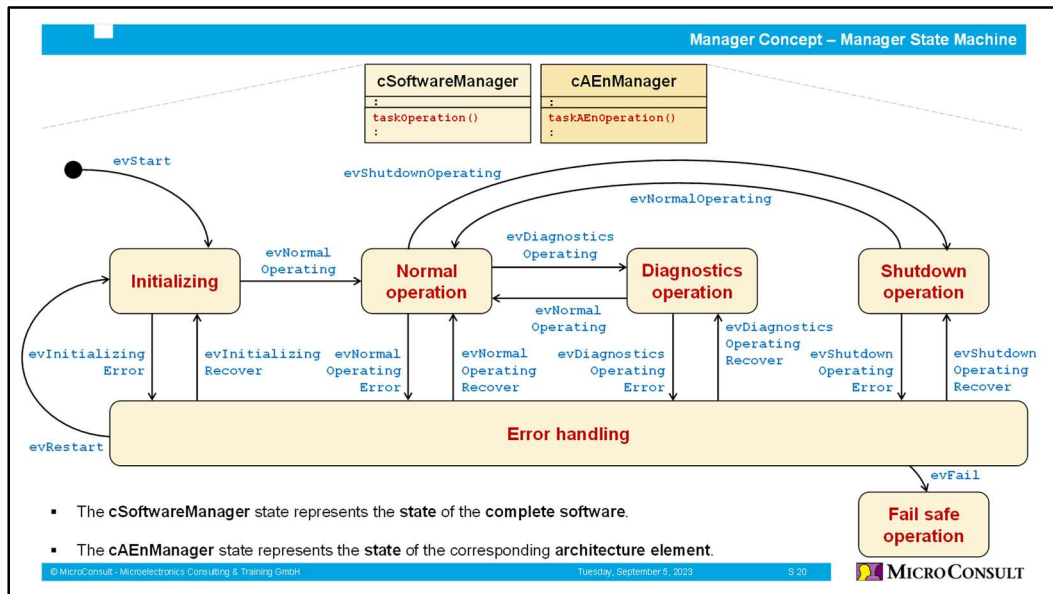
### 2.12.1 Central



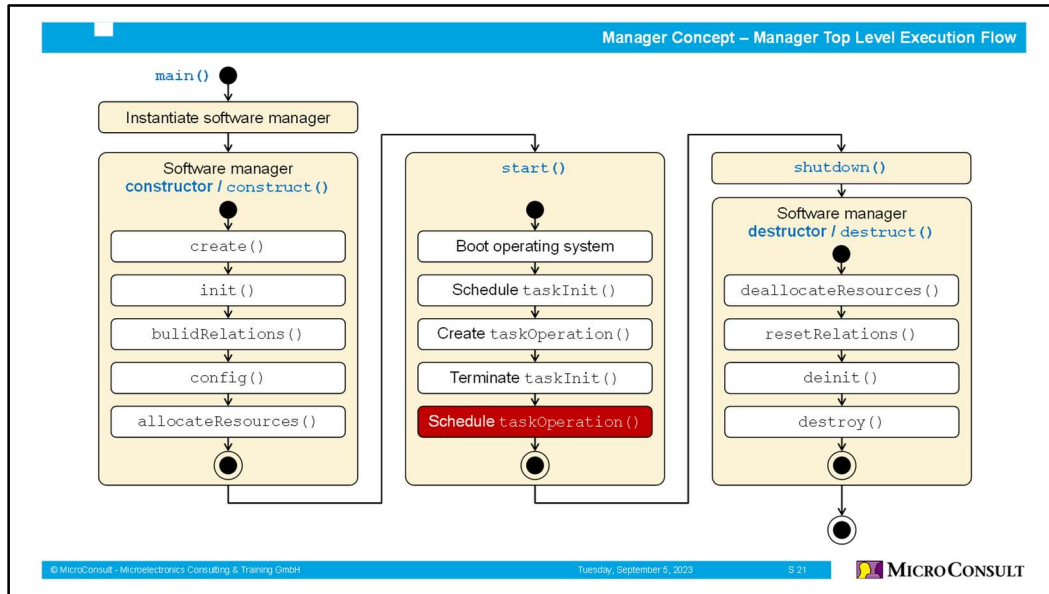
## 2.12.2 Decentral



## 2.13 Manager State Machine



## 2.14 Manager Top Level Execution Flow



## 2.15 Discussion Topics and Improvements


Manager Concept – Discussion Topics and Improvements

- Some missing details:
  - Operation of **error notification** when errors are not polled by the managers.
- The manager classes / modules are not equipped with software **interface(s)**.
- Some design principles are not fulfilled by the manager classes / modules.
  - Principle of magic seven to twelve:  
**Too many functionalities** are located in one class / module.
  - SRP - Single responsibility principle:  
**Different functionalities** are located in one manager class / module.
- Possible approaches:
  - **Add callback interfaces for error notification.** → Refer to the next view.
  - **Add software interfaces.** → Refer to the view after next.
  - **Segregate the manager into different managers according to their functionalities.**
    - Increases design and code complexity tremendously.
    - This approach is **NOT** recommended!

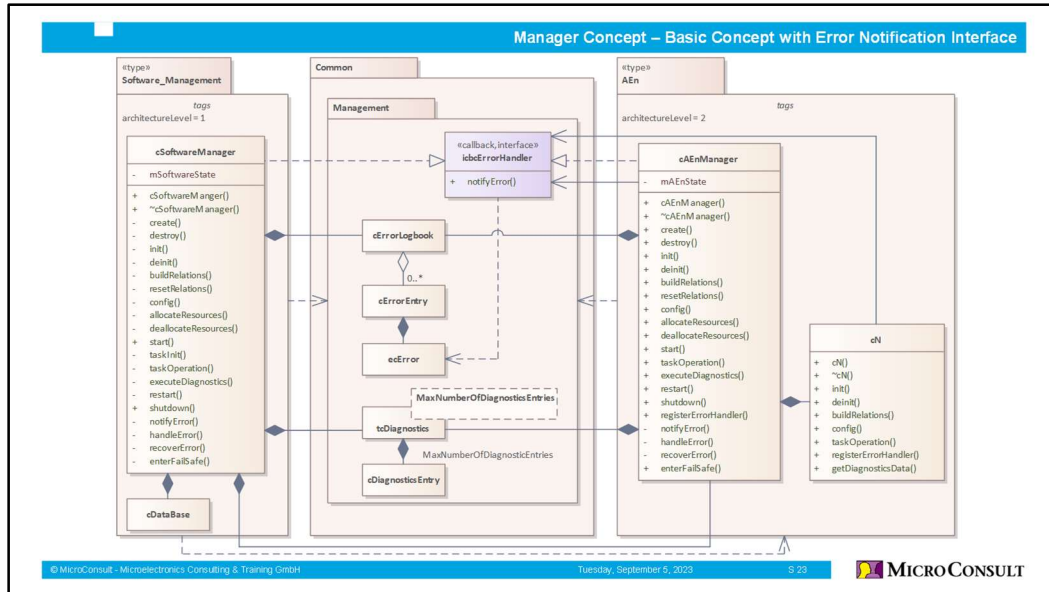
© MicroConsult - Microelectronics Consulting & Training GmbH

Tuesday, September 5, 2023

S.22

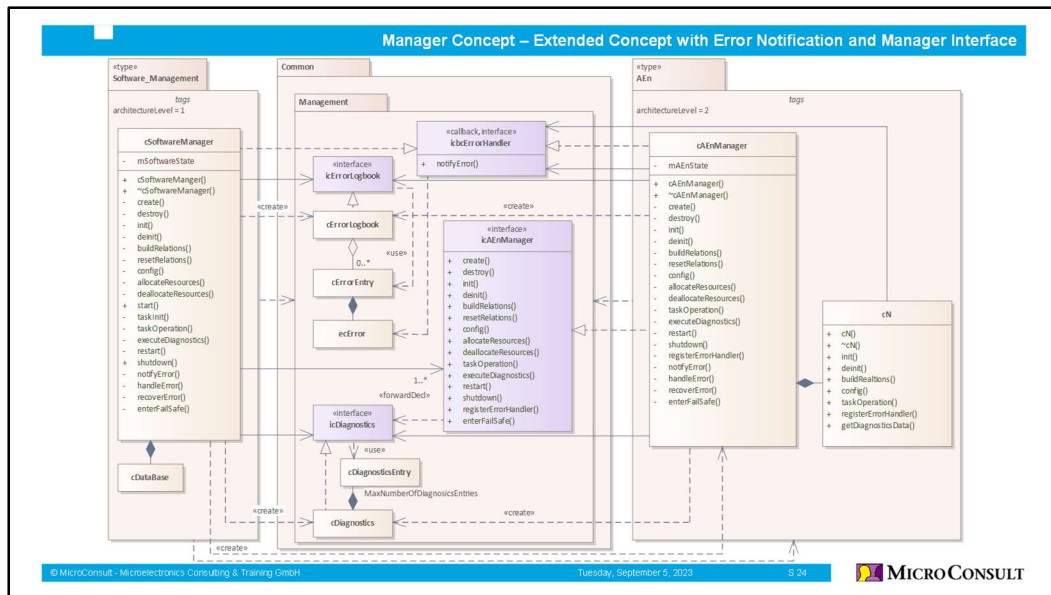
 MICROCONSULT

## 2.15.1 Basic Concept with Error Notification Interface






### 2.15.2 Extended Concept with Error Notification and Manager Interface



### 3 Summary and Outlook

Summary and Outlook

- The purpose of the manager pattern is to **cope** with **software system wide tasks**.
- The manager pattern supports **flexibility**, **scalability** and **expandability** to a great extent.
- Different **detailed design variants** are feasible, depending on:
  - software requirements
  - software architecture
  - programming approach
  - programming language
- This ESE presentation:
  - Download: <http://download.microconsult.net/ese2023/manager-pattern.zip>
- My ESE presentations from the last three years cover similar interesting topics:
  - Download: <http://download.microconsult.net/ese2020/interface-designs.zip>
  - Download: <http://download.microconsult.net/ese2021/port-designs.zip>
  - Download: <http://download.microconsult.net/ese2022/polymorphism.zip>

© MicroConsult - Microelectronics Consulting & Training GmbHTuesday, October 4, 20225/25 MICROCONSULT

## 4 MicroConsult Training and Coaching

MicroConsult Training and Coaching

**Training [English and German]:**

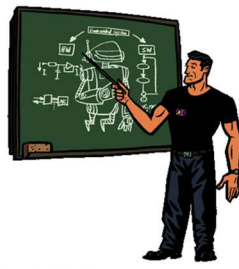
Software Architectures for Embedded and Real-Time Systems  
[English](#)      [German](#)

Embedded C++: Object-Oriented Programming for  $\mu$ C with C++/EC++  
[English](#)      [German](#)

Embedded C++ Advanced: Object-Oriented Programming for  $\mu$ C with C++/EC++  
[English](#)      [German](#)

Embedded Software Design and Patterns with C  
[English](#)      [German](#)

**Coaching [English and German]:**  
[You are welcome to send us your inquiry.](#)



**MicroConsult GmbH**  
 Thomas Batt Dipl.-Ing. (FH)  
 Senior Manager Training & Coaching  
 t.batt@microconsult.com  
 +49 (0)89 450617-35  
 www.microconsult.de

© MicroConsult - Microelectronics Consulting & Training GmbH
Tuesday, October 4, 2022
§ 26

