



Software für Industrie 4.0 (Vorlesung & Übung)

Feldbusse



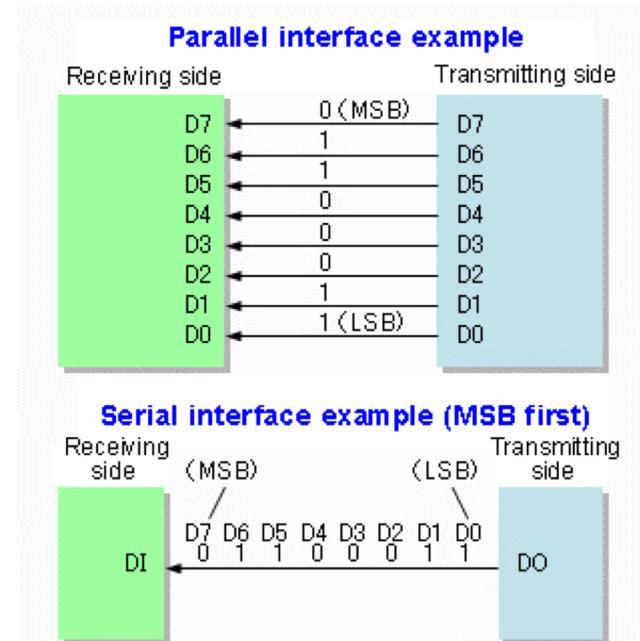
bus

“functional unit for the *transfer of data* between several participants,
these being functional units for data processing, via a *common
transmission path*, wherein *participants are not involved in the transfer* of
data between other participants”

[IEC 60050-351:2006] (Internationales Elektrotechnisches Wörterbuch)

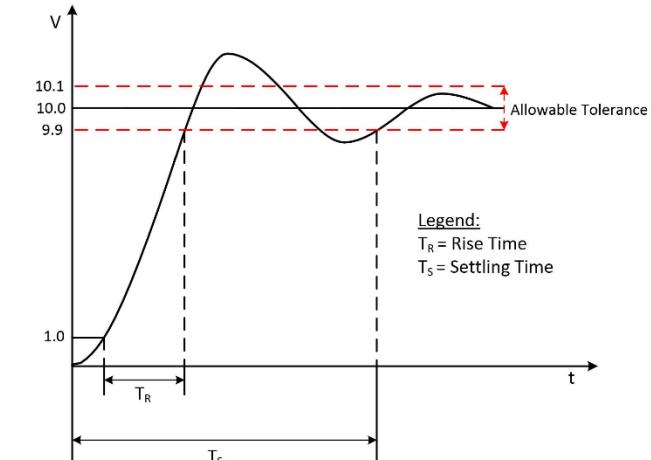
Parallele vs. Serielle Kommunikation

- Beim parallelen Bus existieren üblicherweise drei voneinander getrennte Leitungsgruppen:
 - Adressbus
 - Datenbus
 - Kontrollbus
- Bei der seriellen Übertragung werden diese Informationen hintereinander über eine einzige Leitung versendet

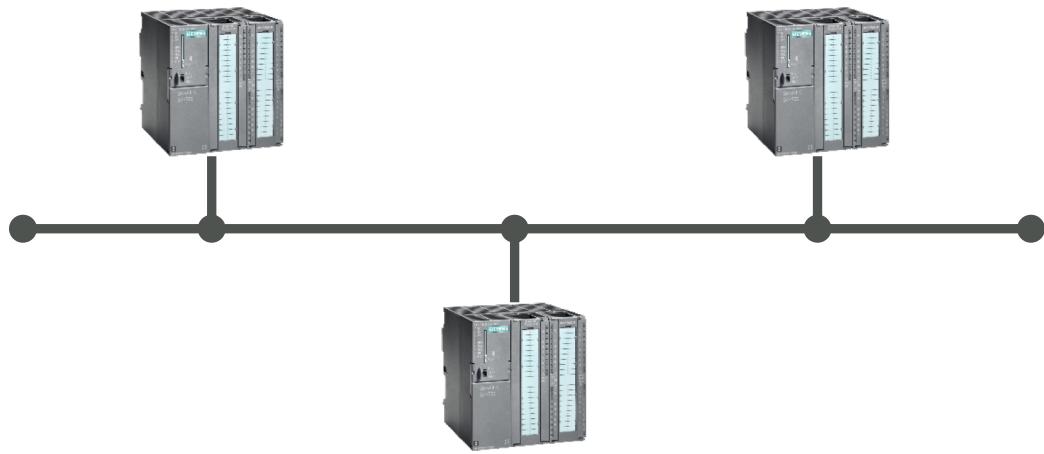


https://upload.wikimedia.org/wikipedia/commons/a/a6/Parallel_and_Serial_Transmission.gif

- Theoretische ist ein paralleles Bussystem deutlich schneller als ein serielles, da ein Byte vs. ein Bit pro Zyklus übertragen werden kann
- Bei hohen **Übertragungsfrequenzen** ist es allerdings deutlich schwieriger die **Übertragungsqualität** eines Parallelbusses aufrecht zu erhalten
 - Crosstalk
 - Settling Time (Zeit bis Bit gelesen werden kann)
- Daher heute bevorzugt serielle Verbindungen mit höheren Frequenzen → bedeutend höhere Transferraten bei einfacherer technischer Umsetzung



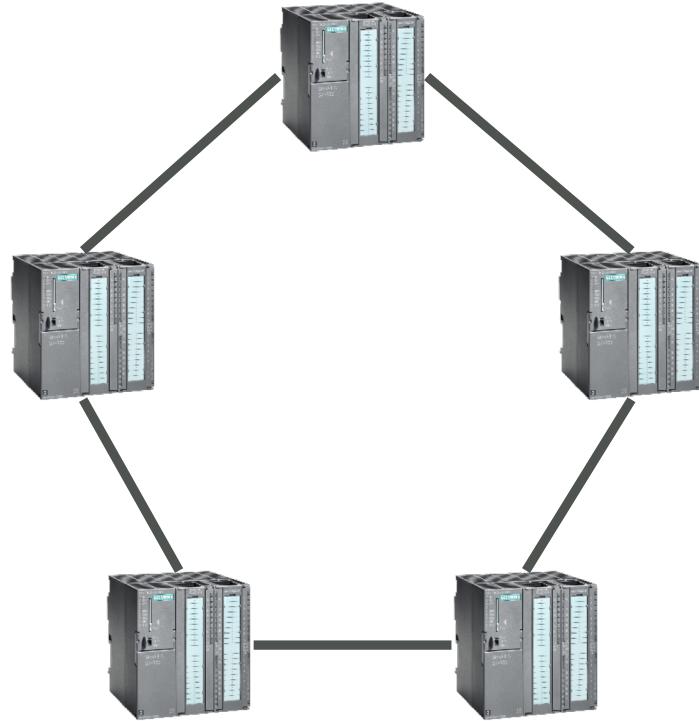
<http://www.ni.com/product-documentation/3104/en/>



Bustopologie (linear)

- + Sehr einfache Verkabelung
(meist nur ein Adernpaar)
- + Oft geringstmögliche Leitungslänge
- + Ausfall eines Gerätes hat keine
Auswirkungen auf die anderen

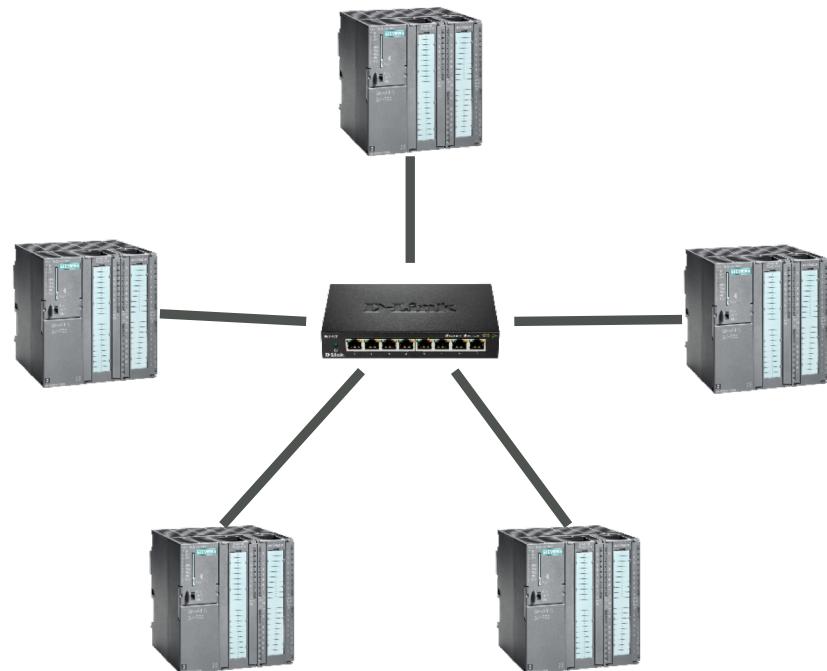
- Bus muss terminiert werden um
Reflexionen an den Kabelenden zu
unterbinden
- Fehlersuche ist durch gemeinsames
Medium erschwert
- Ausfall der einzigen Leitung ist Totalausfall
- Effizienz schwindet mit jedem zusätzlichen
Teilnehmer



Ringtopologie

- + Relativ leicht erweiterbar
- + Nur eine zusätzliche Leitung
- + Nachrichten müssen nicht zurück übertragen werden, sondern im Kreis weiter (+ lange Wege, - kurze Wege)
- + Geringer Zuwachs der Kabellänge durch eine neue Station
- + Höhere Übertragungsraten

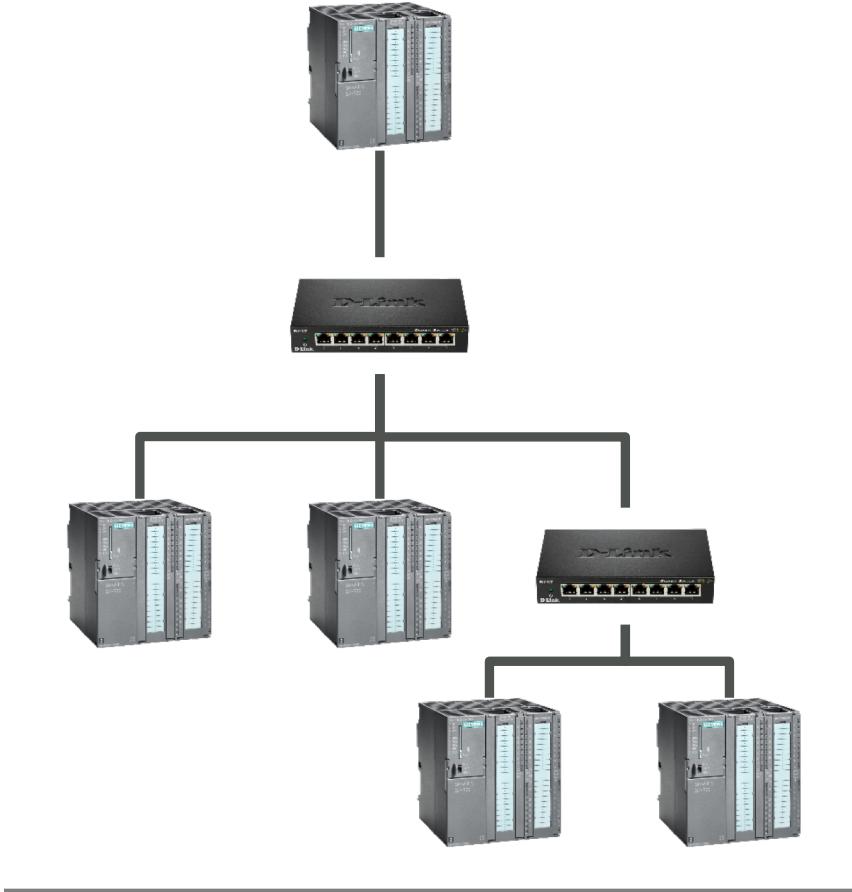
- Bei Ausfall einer Station, üblicherweise Totalausfall des Netzes
- Latenz ist abhängig von der Anzahl der Stationen (Jede Station muss zumindest den Adressierungsbereich erst lesen)



Sterntopologie

- + Stabil, auch bei Kabelbrüchen, etc. ist nur ein Gerät betroffen
- + Daher auch einfache Fehlersuche
- + Punkt-zu-Punkt Verkabelung
- + Einfache Erweiterbarkeit
- + Hohe Übertragungsraten möglich
- + Ausfall eines Gerätes hat keine Auswirkungen auf die Funktion der anderen

- Aufwändiger Verkabelung
- Bei Verteilerausfall ist das gesamte System betroffen



Baumtopologie

- + Ausfall eines Gerätes hat keine Auswirkungen auf die Funktion der anderen
- + Größere Entfernungen realisierbar
- + Durch Verteilerknoten können die Vorteile verschiedener Bustopologien miteinander vereint werden
(dies ist auch in der Praxis sehr üblich)
- Der Ausfall eines Verteilers hat die Unerreichbarkeit des gesamten ihm untergeordneten Teilbaums zur Folge
- Bei großen Bäumen kann es durch die Anzahl der Verteilerknoten zu sehr hohen Latenzen kommen
- Bei Nachrichten zur Wurzel (oder höchsten Ebene) kann es leicht zu Engpässen kommen

- **Strobe**

Master sendet Nachricht an alle Slaves, welche mit Ihren aktuellen Status- und Sensorwerten antworten

- **Polling**

Master sendet an jeden Slave eine eigene Anfrage, welcher mit seinen aktuellen Werten antwortet

- **Cyclic Communication**

Alle Slaves senden periodisch ihre aktuellen Daten an den Master. Teilweise kann der Master auch definieren welche Daten er zyklisch bekommen möchte

- **Sporadic / Acyclic Communication**

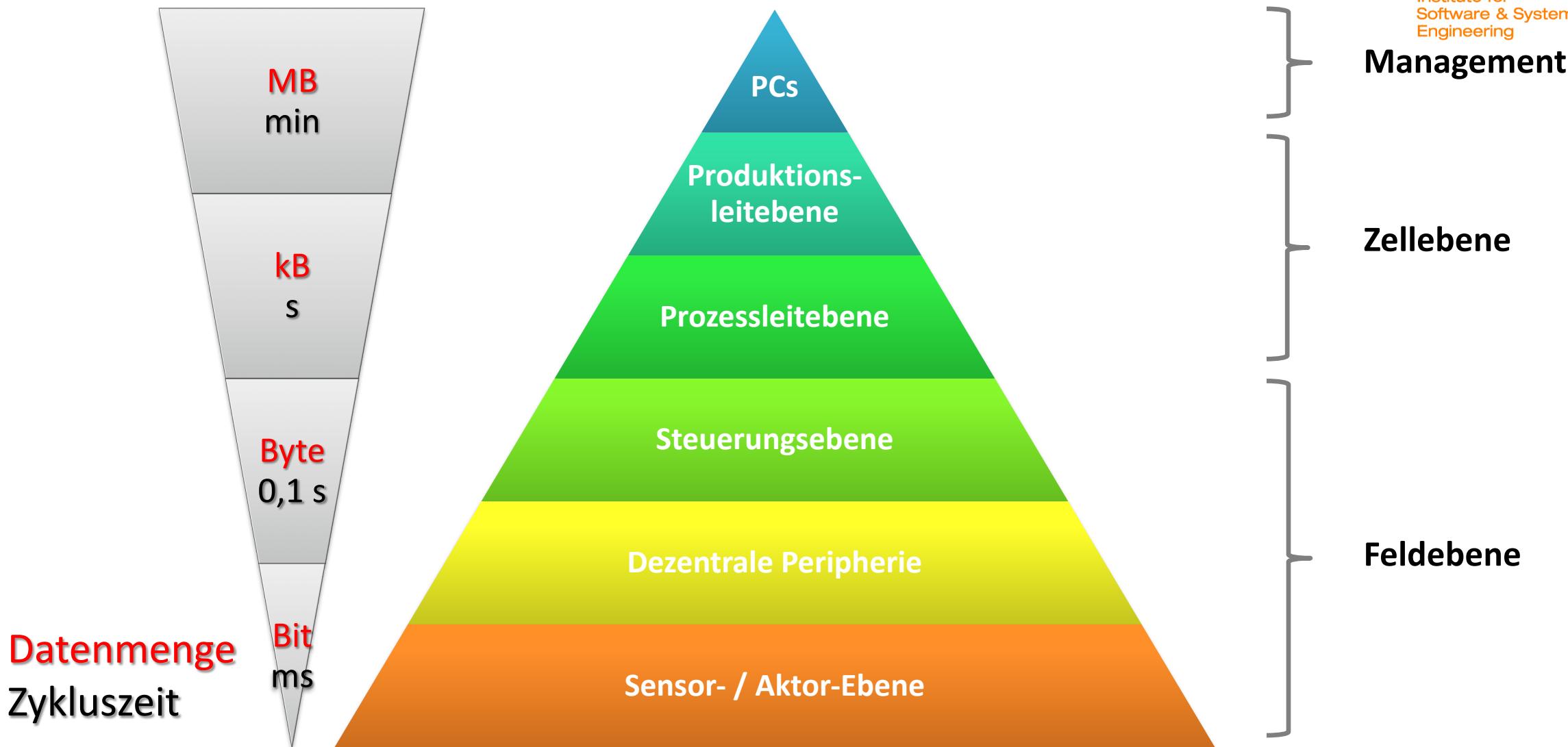
Der Master fragt seine Slaves nur nach neuen Daten wenn er diese braucht (azyklisches Polling)

- **Event-based Communication**

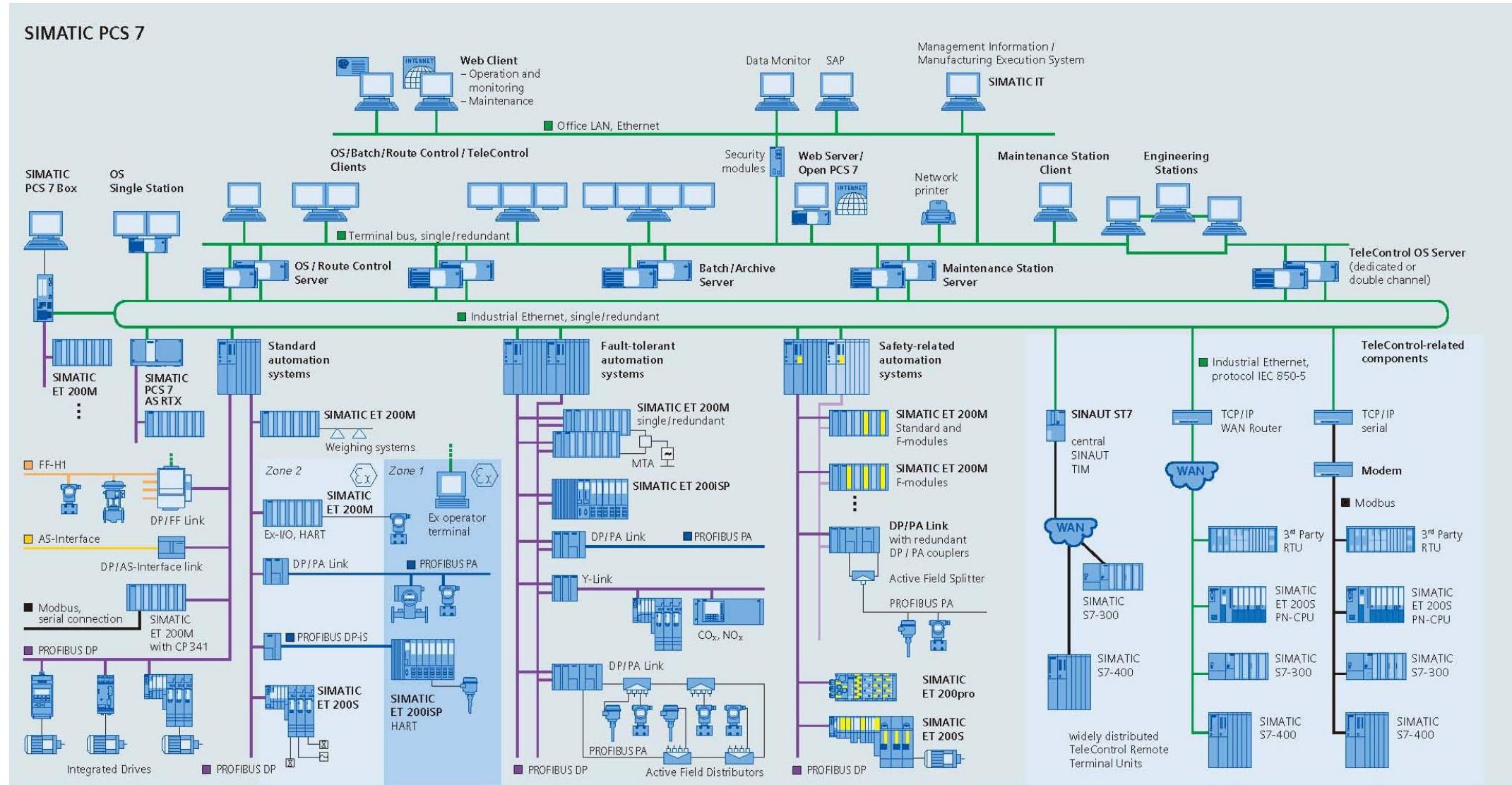
Die Slaves senden dem Master ihre Daten sobald neue vorliegen, also irgendein Ereignis eingetreten ist

- Einfache Installation, Inbetriebnahme & Wartung
- Kosteneffizienz
- Flexibilität in der Topologieauswahl
 - Universelle Topologie (alle Geräte in der gleichen Topologie)
 - Über Gateways/Kopplungsmodule Verbindung von verschiedenen Topologien
- Echtzeitfähigkeit und Zykluszeiten von einigen wenigen Millisekunden

Feldbusse in der Industrie



Bustopologien – Kombinationen in der Baumtopologie



ÜBERSICHT VERSCHIEDENER FELDBUSSE

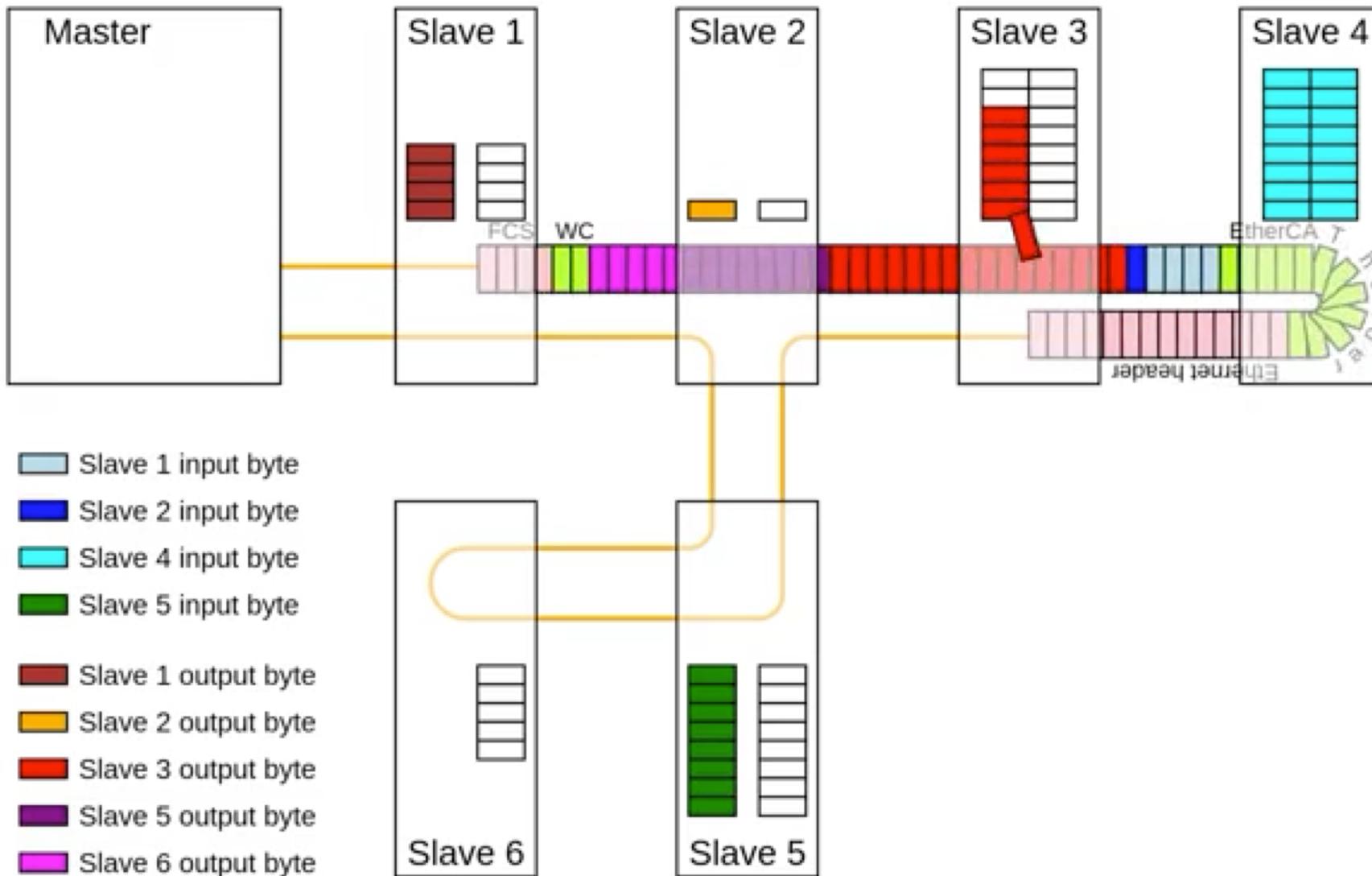
Foundation Fieldbus (FF)



- Ursprünglich als Ablösung des 4-20mA Systems konzipiert
- Einführung des H1-Standards im Jahr 1996 (32 kbit/s)
- Stellt wie 4-20mA Strom und Kommunikation über ein Leitungspaar bereit
- HSE-Standard (2000) nutzt High-Speed-Ethernet (100 Mbit/s) und kann im Moment noch keine Stromversorgung liefern (Power-over-Ethernet, kurz PoE soll aber integriert werden)
- Immer noch ein weit verbreitetes Bussystem, insbesondere in der Automatisierung von weniger komplexen Systemen, wie Trinkwasser- oder Energieversorgung

- EtherCAT (Ethernet for Control Automation Technology) von Beckhoff
- Anders als bei anderen Bussystemen wird bei EtherCAT nicht der gesamte Frame gelesen, verarbeitet und dann eine Antwort verfasst, sondern es werden beim Empfang bereits alle für das Gerät bestimmten Daten eingelesen, und die benötigten Antworten eingefügt
- Dadurch sind sehr geringe Verzögerungszeiten pro verbundenem Endgerät möglich
- EtherCAT verwendet als physikalisches Medium Ethernet-Verbindungen, sowie das Standard-Ethernetprotokoll

EtherCAT



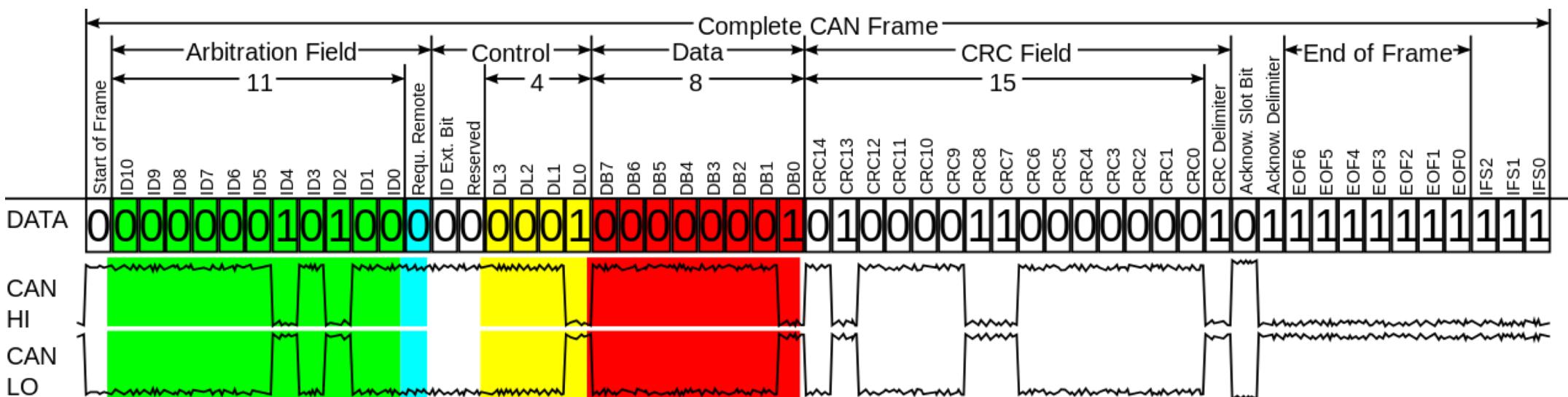


- 1983 von Bosch für die Automobilindustrie entwickelt
- Ziel war es, die Komplexität des zentralen Kabelbaums zu reduzieren, u.A. um Gewicht einzusparen
- International standardisierter Bus als ISO 11898
- Heutzutage auch als Bussystem in der Automatisierung stark verbreitet
- Topologie: typischerweise lineare Topologie, optional mit kurzen Stichleitungen; auch Stern-Topologie möglich
- 2 verschiedene Ausführungen von CAN:
 - Lowspeed CAN: 125 kbit/s
 - Highspeed CAN: 1 Mbit/s
- Allerdings unterscheiden sich Lowspeed und Highspeed noch in weiteren Eigenschaften, wie beispielsweise den Definitionen für HIGH- und LOW-Pegel, sodass sie nicht miteinander kompatibel sind

CAN – Signalverlauf am Bus

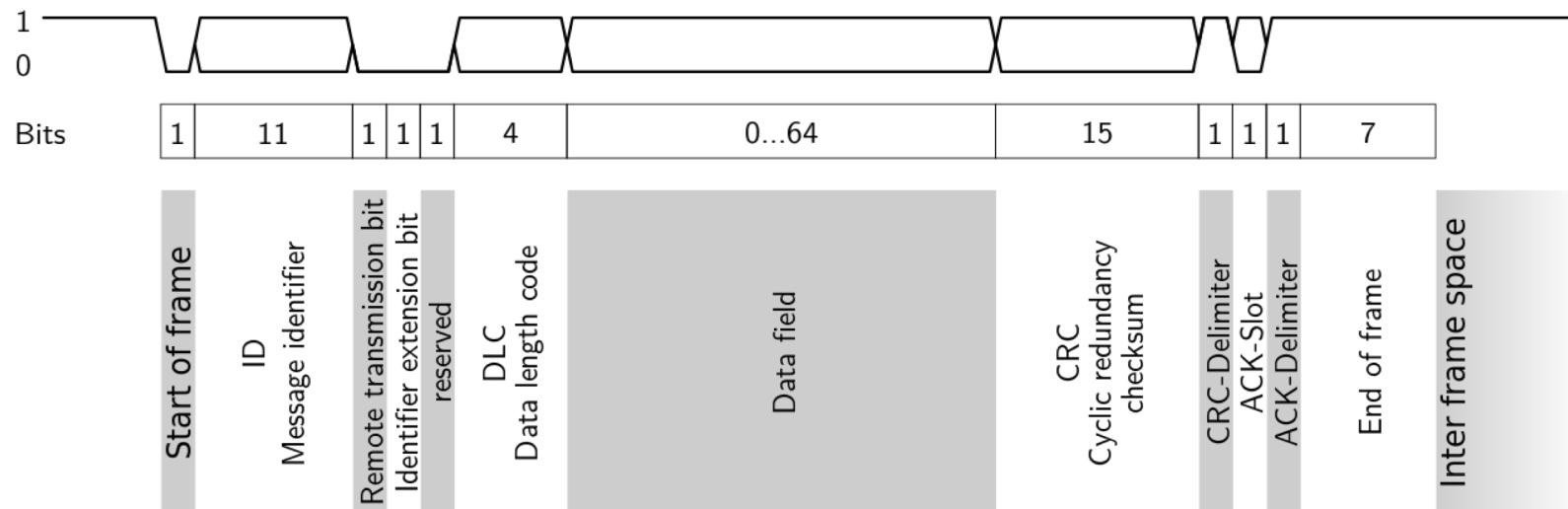


- Da Rauschen und Zeitverzögerungen beim Pegelwechsel auf physikalischen Gegebenheiten beruhen, müssen diese kompensiert werden
 - Da nur digitale Signale benötigt werden, können Korridore festgelegt werden, in denen die logischen Pegel angesiedelt sind
 - Zudem synchronisieren sich die Busteilnehmer auf die Frequenz der ersten Flanken, um so jedes Bit auch zum richtigen Zeitpunkt lesen zu können



CAN – Aufbau eines Frames

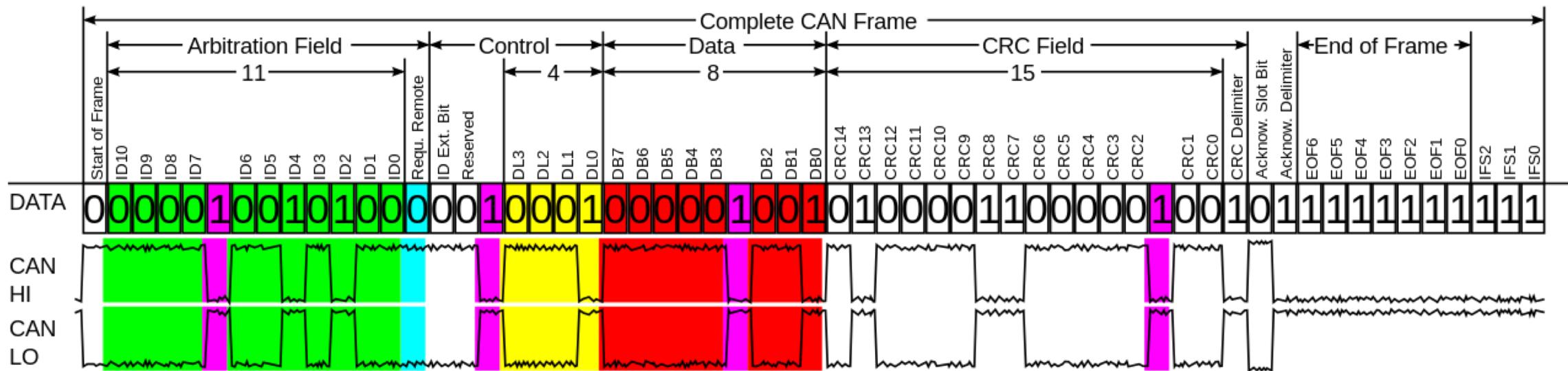
- Bei CAN werden keine konkreten Geräte adressiert, sondern Datenobjekte
- Zur Unterscheidung mehrerer gleicher Geräte an einem Bus kann meist ein Offset für das konkrete Datenobjekt definiert werden
- Mit dem „Remote Transmission Bit“ wird angegeben, dass es sich um eine Abfrage von Daten von einem Slave handelt, weshalb eine direkte Antwort zu erwarten ist
- Im ACK-Feld bestätigen die Slaves (mit dominantem Bit) dem Master (legt rezessives Bit an) den korrekten Empfang eines CAN-Frames
- Sollte ein Frame nicht korrekt empfangen werden (Slave müsste rezessives Bit anlegen), so muss der betreffende Slave nach dem ACK-Delimiter ein Error-Flag auf den Bus schreiben



CAN – Bitstuffing



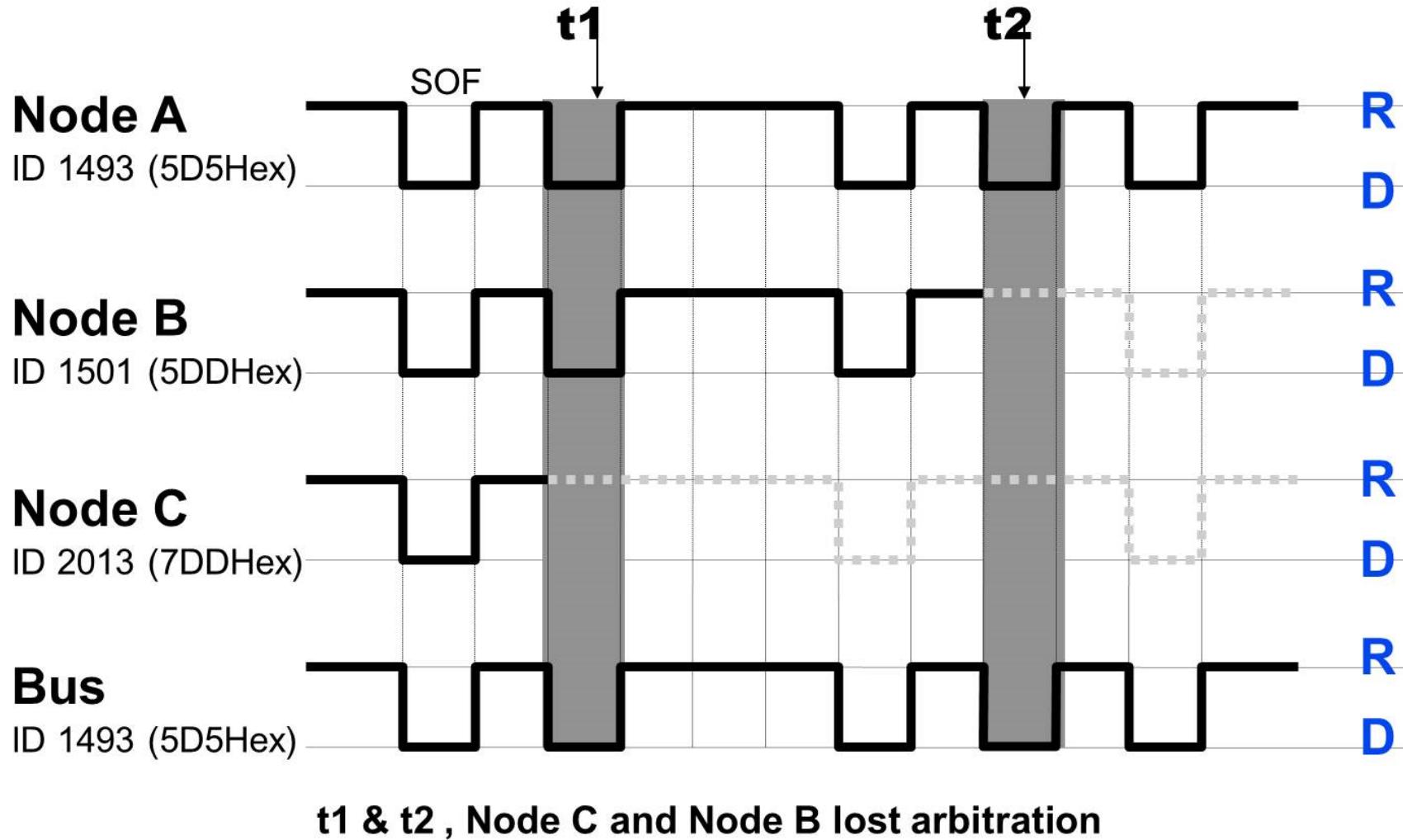
- Da z.B. eine Folge von 7-Bit logischer 1 als Terminierung der Übertragung (End-of-Frame) interpretiert wird, muss Bitstuffing verwendet werden, um diese Semantik nicht mitten in der Übertragung zu erzeugen
- Bei CAN wird nach 5 Bits mit gleichem Pegel ein entgegengesetztes Bit geschrieben
- Dies verlängert den Frame indeterministisch
- Die Stopfbits werden außerdem zur Nachsynchronisierung der Teilnehmer während der Übertragung verwendet





- Elektrisch gesehen gibt es auf dem CAN-Bus ein rezessives und ein dominantes Bit
 - Rezessiv: Die logische 0 (LOW- oder GND-Pegel bei CAN HIGH) liegt ohne explizite Aktion am Bus an
 - Dominant: , Die logische 1 (HIGH-Pegel) muss vom Kommunikationsmodul am Bus angelegt werden (mit GND verbunden)
- Hierdurch kann eine Arbitrierung auf dem Bus realisiert werden, indem jeder Busteilnehmer bei jedem Bit überprüft, ob sein rezessives Bit tatsächlich am Bus anliegt
- Andernfalls muss er seine Übertragung abbrechen, da sonst die Daten nicht rekonstruierbar sind
- Dies hat zudem den Nebeneffekt, dass höherpriore Nachrichten die Arbitrierung immer gewinnen (ID 0 als höchste Priorität: nur dominante Bits)
- Auf der CAN LOW Leitung ist die Dominanz, wie die logische Bedeutung der Pegel, invertiert

CAN – Arbitrierung



- **CAN-FD (Flexible Datarate):**

Erlaubt eine höhere Datenrate im Datenbereich der Nachricht.
Außerdem ist die maximale Nachrichtenlänge von 8 auf 64 Byte
erweitert worden

- **CANopen (EN 50325-4):**

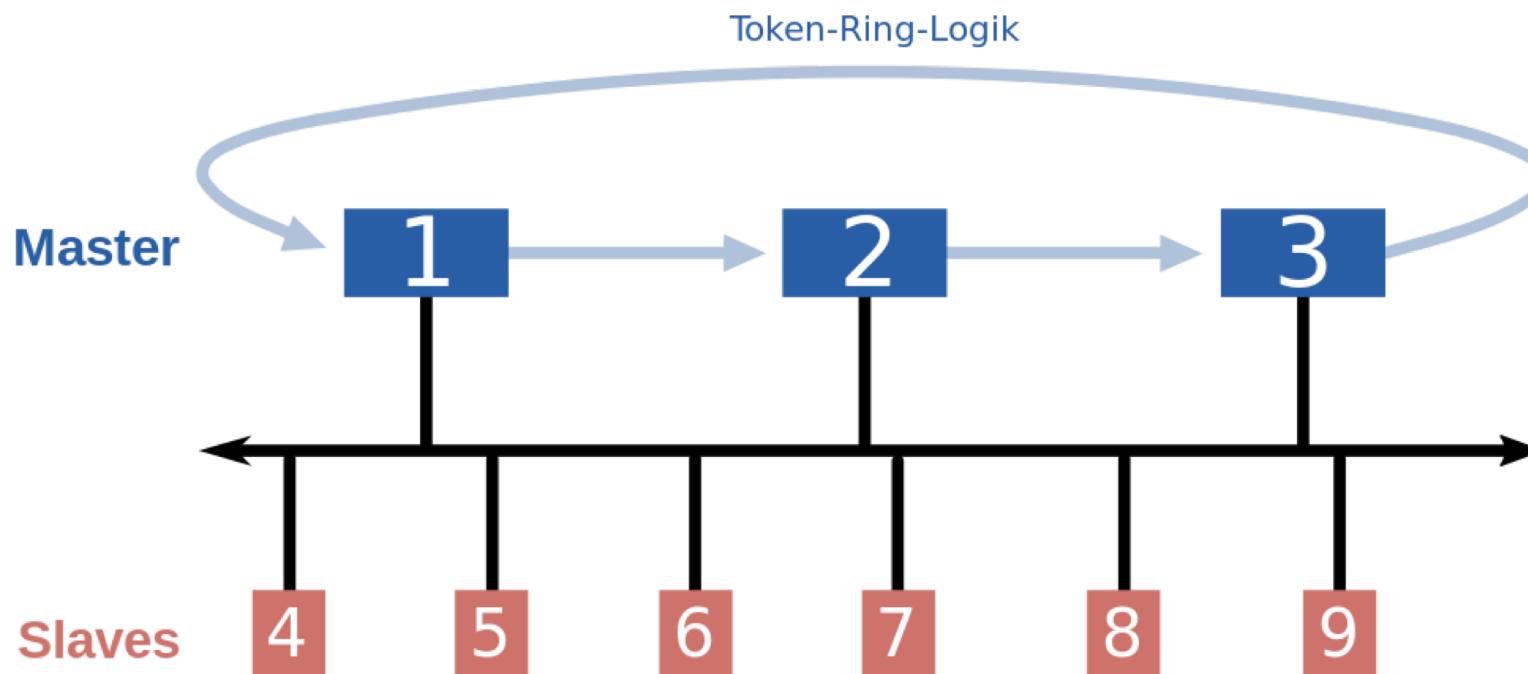
- Eindeutige Identifizierung von Geräten
- *Servicedatenobjekte (SDO)*: Hiermit werden Parametrisierungen vorgenommen (azyklische Übertragung)
- *Prozessdatenobjekte (PDO)*: Hierüber wird der zyklische Datenaustausch realisiert. Durch vorherige Konfiguration der Einträge im PDO (→ SDO) wird der Overhead reduziert. Es stehen nur die Rohdaten im Paket.
- *Netzwerkmanagement-Objekte (NMT)*: Hiermit wird der Gerätezustand bestimmt und überwacht
- CANopen übernimmt des Weiteren die Aufteilung langer Nachrichten in mehrere CAN-Frames (nicht PDOs!)
- Durch neue Nachrichten-IDs können in CANopen 65536×254 Datenobjekte unterschieden werden (11-Bit CAN-ID + 16-Bit CANopen Objekt-ID)





- **PROFIbus (Process Field Bus)** wurde ab 1987 von einem deutschen Konsortium aus Firmen und Forschungseinrichtungen entwickelt
- Direkter Konkurrent zu FOUNDATION Fieldbus
- Heute der am weitesten verbreitete Feldbus-Standard mit 43,8 Mio im Einsatz befindlichen Geräten
- Erste Spezifikation: **PROFIbus FMS** (Fieldbus Message Specification), für sehr komplexe Anwendungsszenarien gedacht und von PROFIbus DP abgelöst worden
- **PROFIbus DP** (Dezentrale Peripherie): Ansteuerung von Sensoren und Aktoren durch eine zentrale Verarbeitungseinheit. Die tatsächliche Ansteuerung der Aktoren und Auswertung der Sensoren erfolgt allerdings im Feld, was zu einem deutlich geringeren Verkabelungsaufwand führt
- **PROFIbus PA** (Prozessautomation): ähnlich wie 4-20mA wird mit Hilfe der **Manchester Bus Powered (MBP)** Technik die Datenübertragung und Stromversorgung mit einem Adernpaar realisiert

- Durch einen Standard-Bus kombiniert mit einem Token-Ring Netzwerk für die Kommunikation von Master-Geräten untereinander, kann ein Multi-Master-System ohne Bus-Arbitrierung realisiert werden
- Nur derjenige Master, der aktuell das Token hält, darf auf den Bus zugreifen

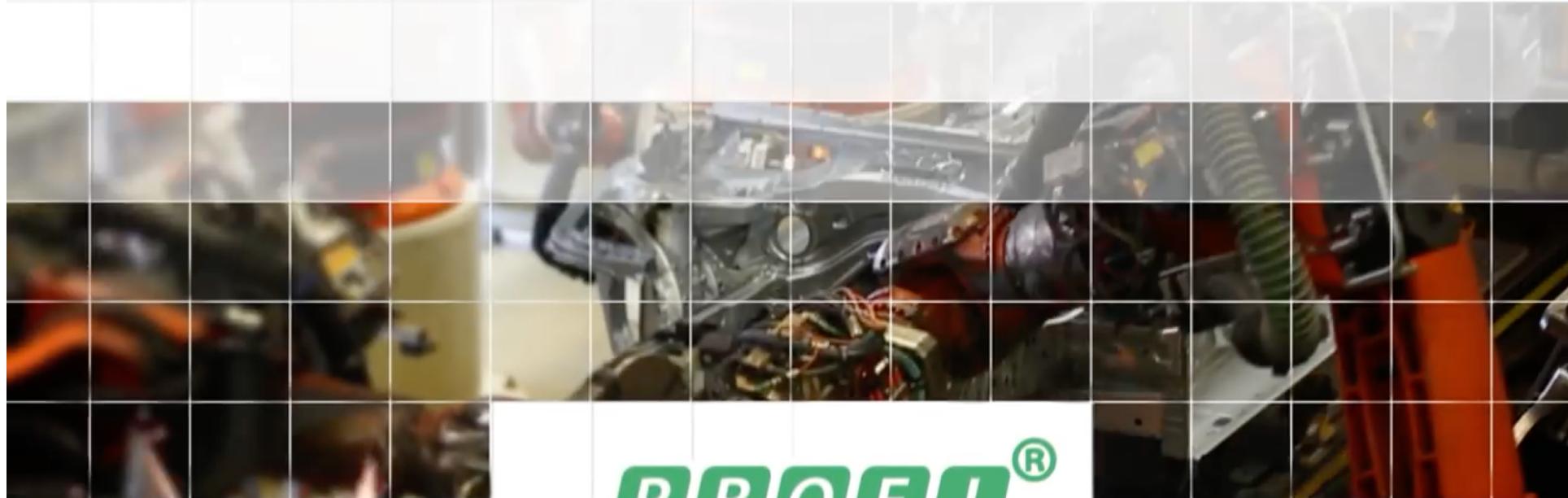




- Zur Vereinfachung und Standardisierung typischer Applikationsszenarien im Ökosystem von PROFIbus, wurden einige spezifische Profile entwickelt
- **PROFIdrive:** Für alle Automatisierungsaufgaben mit Antrieben (eins der häufigsten Szenarien). Wurde zusammen mit den Herstellern von Antrieben definiert und abstrahiert die konkrete Ansteuerung eines Antriebes über ein Interface.
- **PROFIsafe** zur einfacheren Implementierung von funktionalen Sicherheitsanforderungen (Anbindung von Not-Stop, Lichtschranke, etc.) an die zentrale Steuerung über ein beliebiges anderes Bussystem mit garantierten Sicherheitseigenschaften (sonst zusätzliches Netzwerk und Kommunikationsschnittstellen nötig)
- **PROFIenergy** vereinfacht die prozessweite Energieoptimierung



PROFINET – Der Film

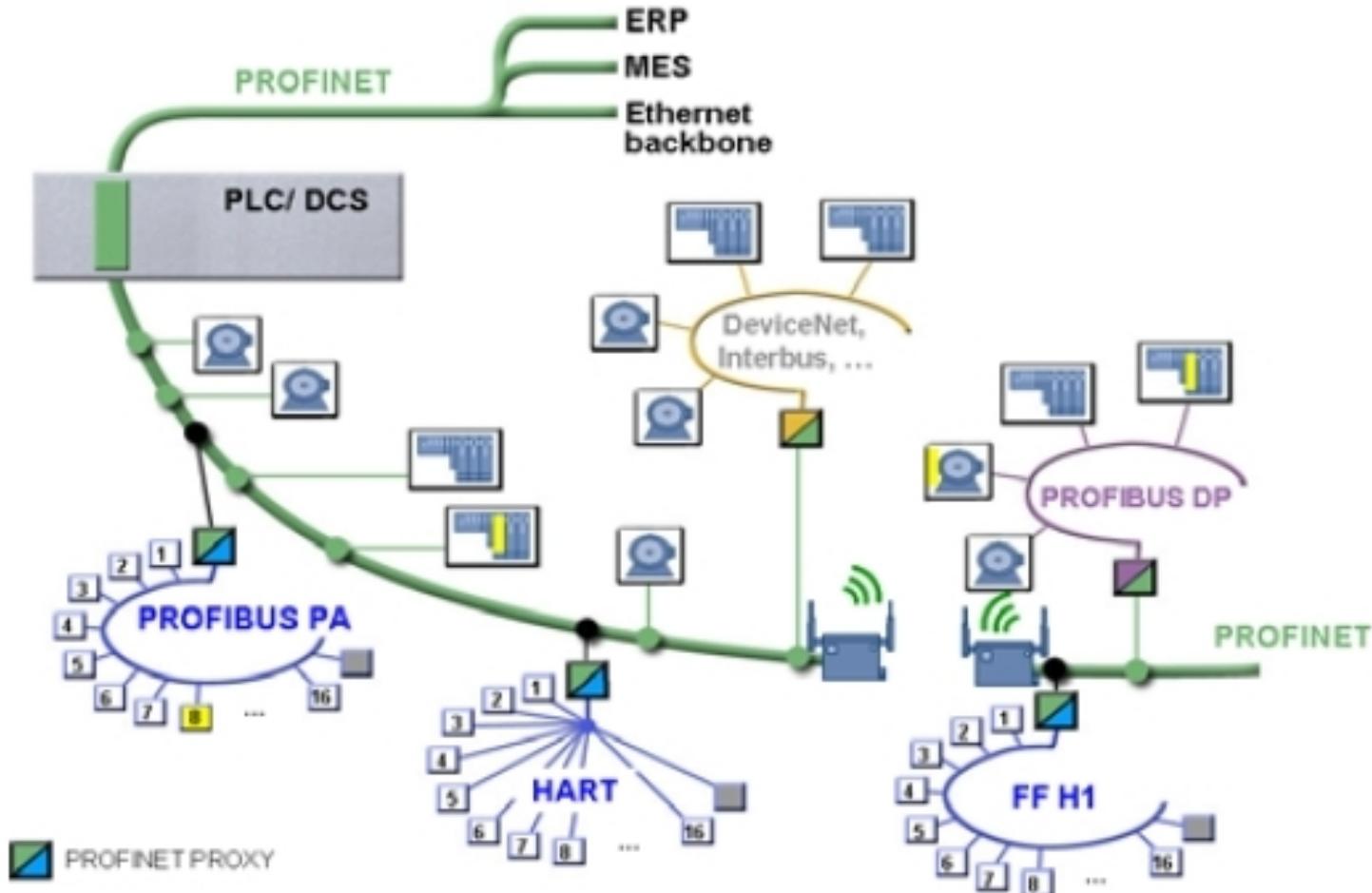


PROFINET®



- PROFINet ist eine Netzwerklösung für Automatisierungsaufgaben, die auf Standard-Hardware von Ethernet aufbaut
- Für nicht-echzeitkritische Daten wird der Standard TCP/IP-Stack verwendet
- Für echtzeitkritische Daten stehen die Profile PROFINet RT (RealTime, 1-10ms) und PROFINet IRT (Isochronous RealTime, unter 1ms) zur Verfügung, welche aufgrund der speziellen Anforderungen den TCP/IP-Stack durch einen proprietären ersetzen
- Aufgrund dieser Eigenschaften kann PROFINet auch mit Standard-Ethernet-Geräten auf einem Bus koexistieren, für Interoperabilität sind allerdings Protokollübersetzer erforderlich

PROFINet - Topologie





- PROFINet unterstützt die selben Protokollerweiterungen und Applikationsprofile wie PROFibus
- Zusätzlich existiert **PROFINet IO**, welches die Funktionalität von PROFibus DP abbildet und folgende Geräteklassen definiert:
 - **IO-Controller:** Typischerweise eine SPS, auf welcher das Automatisierungsprogramm läuft
 - **IO-Supervisor:** Ein Programmier-, Diagnose-, oder Bediengerät welches keine echtzeitkritischen Daten benötigt
 - **IO-Device:** Die Entsprechung eines PROFibus-Slaves, also das tatsächliche IO-Gerät, welches die Ansteuerung von Sensoren & Aktoren im Feld übernimmt

Empfohlene und weiterführende Literatur

- [Fieldbus:2001] „The FOUNDATION Fieldbus primer“
<http://www.fieldbusinc.com/info/primer.html>
- [Wittgruber:2002] „Digitale Schnittstellen und Bussysteme: Einführung für das technische Studium“, Springer Verlag, 2002
- [CANopen] „CAN in Automation (CIA)“-Organisation <http://www.can-cia.org/can-knowledge/canopen/canopen/>
- [Feldbusse.de] Überblick über viele aktuell verwendete Bustechnologien, inkl. PROFINet & PROFIBus <http://www.feldbusse.de/index.asp>
- [PROFINet.com] Überblick über PROFINet (offizielle Seite des PROFINet Konsortiums) <http://us.profinet.com/technology/profinet/>