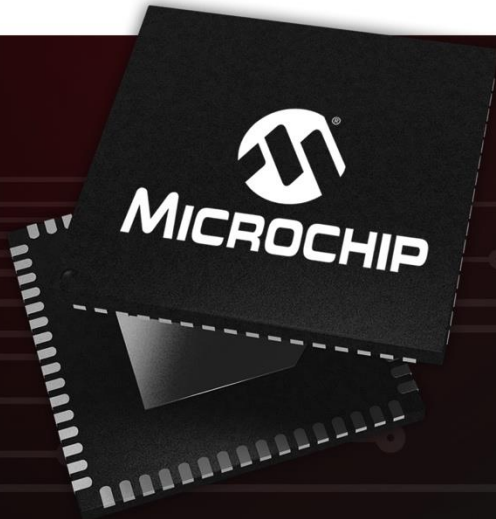




# MICROCHIP



A Leading Provider of Microcontroller, Security,  
Mixed-Signal, Analog & Flash-IP Solutions



***Migration of a classic infotainment network to Linux/AGL***  
***Roland Trissl, Senior Applications Engineer***  
***June 2018***

# Agenda

---

- **Classic infotainment network (MOST®)**
- **Migration to Linux-based MOST**
- **Migration to AGL-based UNICENS**
- **Migration to AGL-based MOST**



**MICROCHIP**

# **Classic infotainment network (MOST<sup>®</sup>)**



- **Media Oriented Systems Transport (MOST) is:**
  - A synchronous network based on Microchip's Intelligent Network Interface Controller (INIC) network controllers
  - Uses Microchip's MOST NetServices library (MNS) to communicate with INIC inside every device
  - An application layer standard that defines communication (function blocks) and system behaviour

# Data types

**Control data**  
Control Messages  
used for  
Function Blocks

**Streaming Data**  
Synchronous streams  
(PCM audio etc.) or  
Isochronous Streams  
(MPEG video etc.)

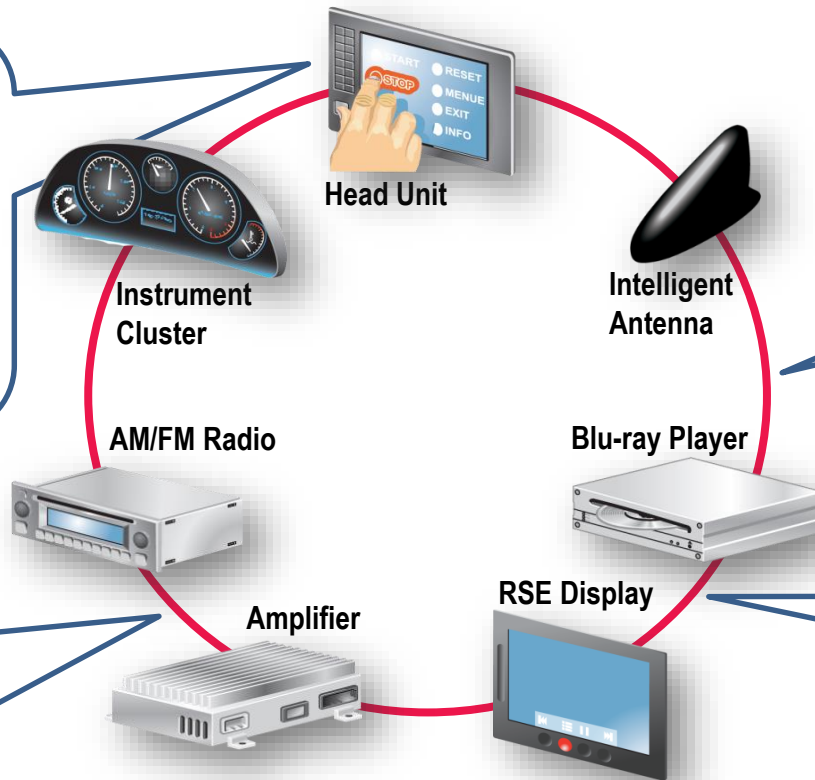
**Packet Data**  
Ethernet packets  
(TCP/IP, UDP, etc.)  
or other packets



# MOST<sup>®</sup> network

**System Setup** and establishing of **Streams** via *NetworkMaster* and *ConnectionMaster* functionality (MSMM) in Head Unit and Functions in each device

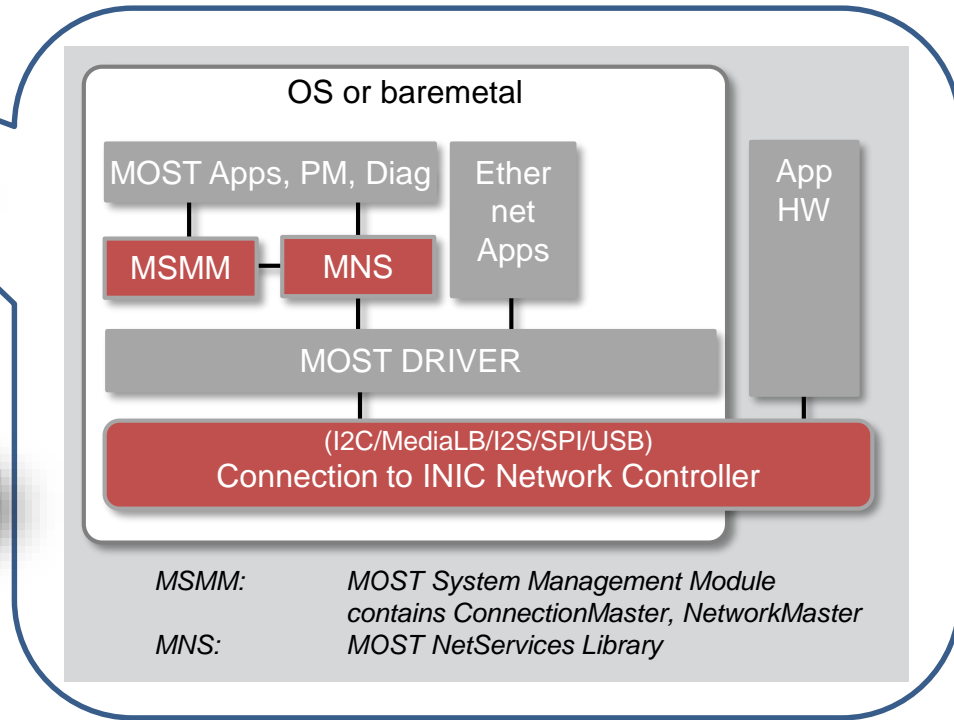
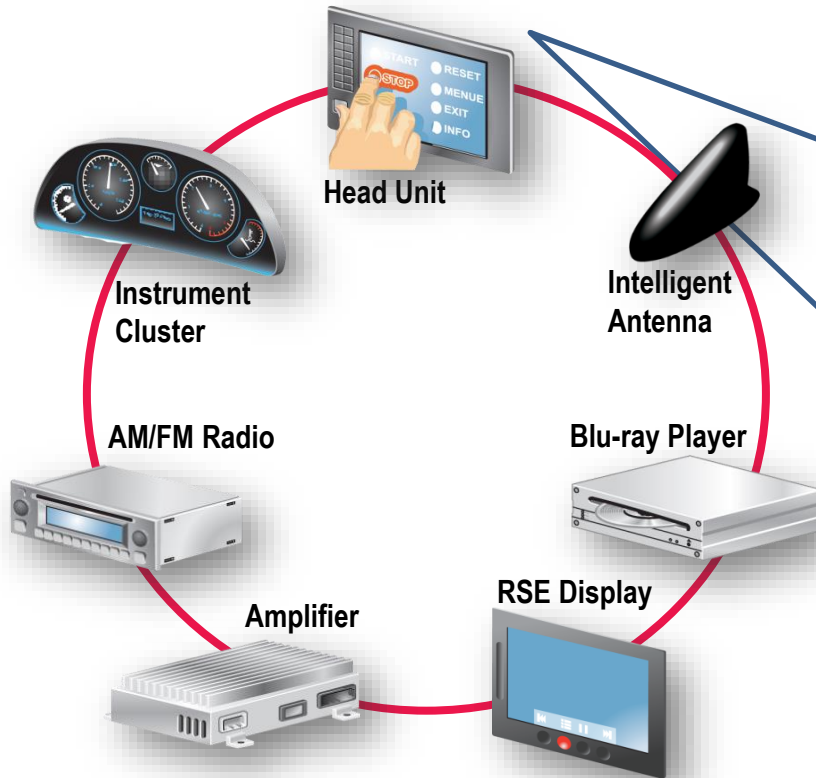
Each node contains *MOST NetServices* SW stack to communicate with network controller



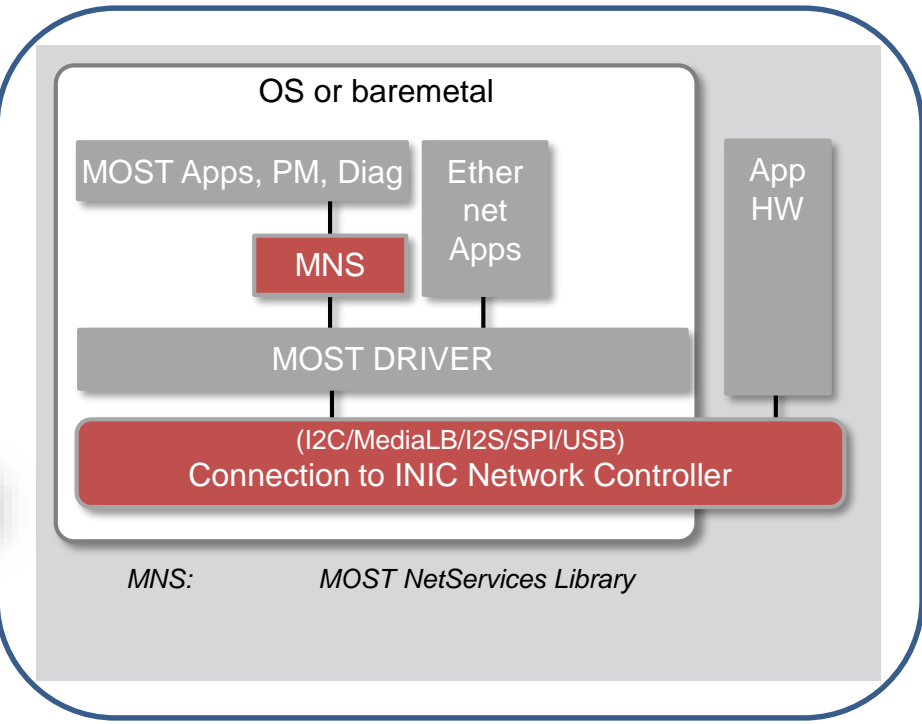
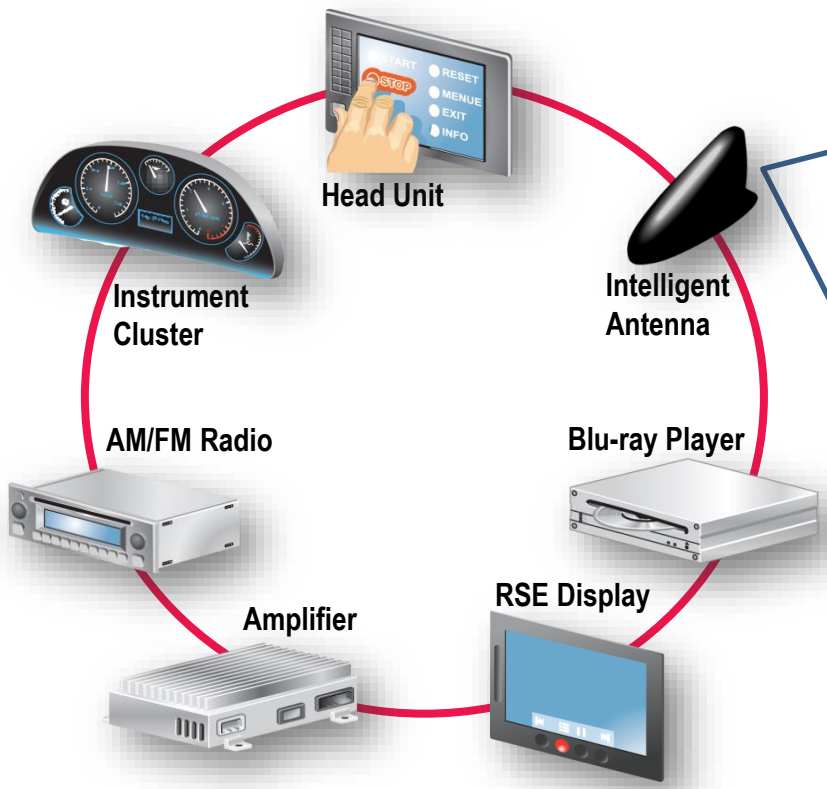
Communication between devices is using *Functions* implemented in each device

**Device Control** via Functions

# Master node



# Slave node





# NW functionality/interfaces

---

## System Setup

NetworkMaster  
ConnectionMaster  
MOST App  
FBlocks

PowerMgmt  
Diagnosis

## Device Control

MOST App  
FBlocks

Ethernet Apps

## Streaming

I<sup>2</sup>S  
MediaLB  
USB  
via Driver

App HW



**MICROCHIP**

**Migration from  
Classic MOST<sup>®</sup>  
to  
Linux-based MOST**

---

# Motivation

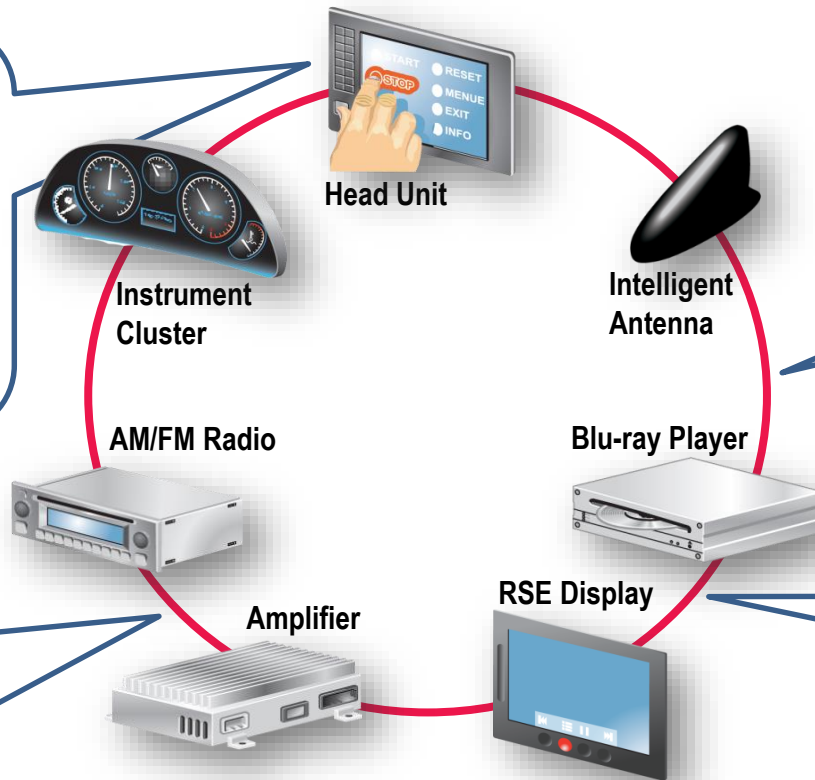
---

- **Replace single devices with Linux devices**
  - Take advantage of Linux benefits
  - No change in network behaviour
  - Reuse as much MOST<sup>®</sup> App code as possible
- **Keep some existing devices unchanged**

# MOST<sup>®</sup> network

**System Setup** and establishing of **Streams** via *NetworkMaster* and *ConnectionMaster* functionality (MSMM) in Head Unit and Functions in each device

Each node contains *NetServices* SW stack to communicate with network controller

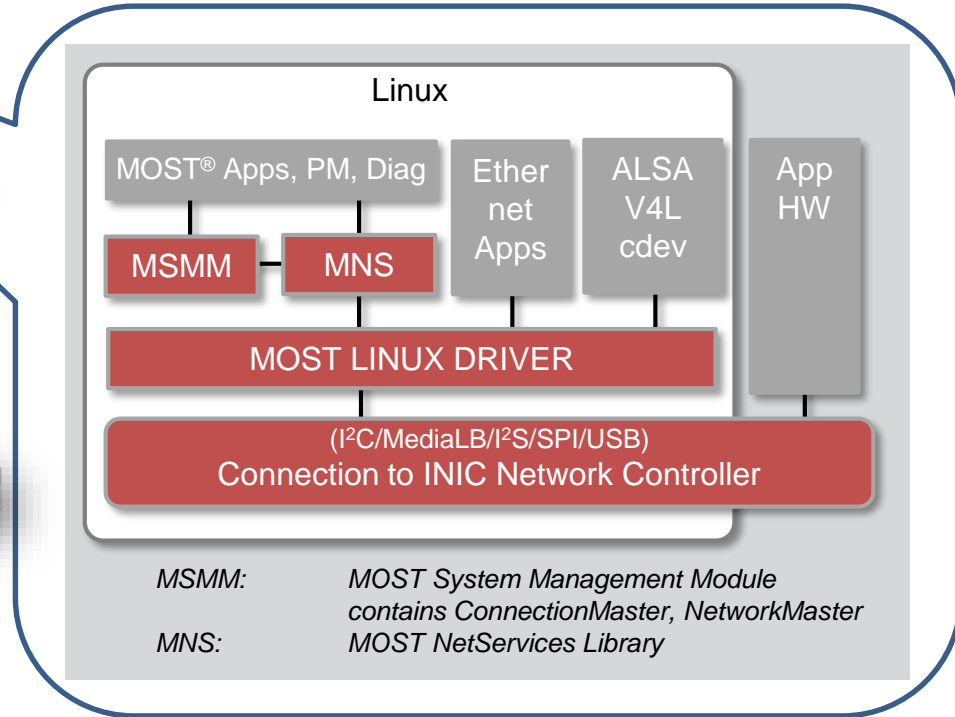
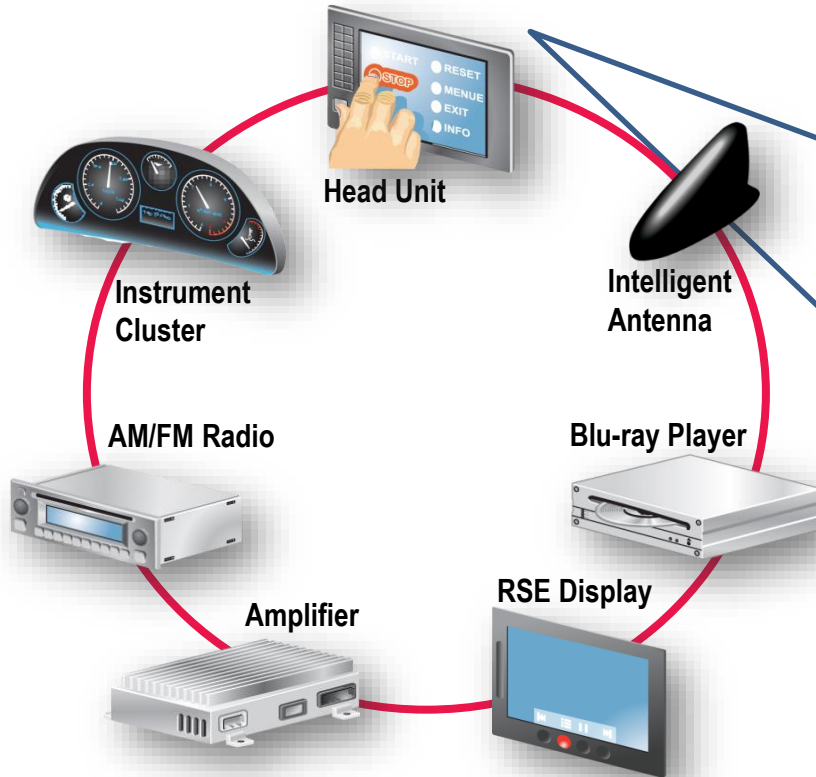


Communication between devices is using App Functions implemented in each device

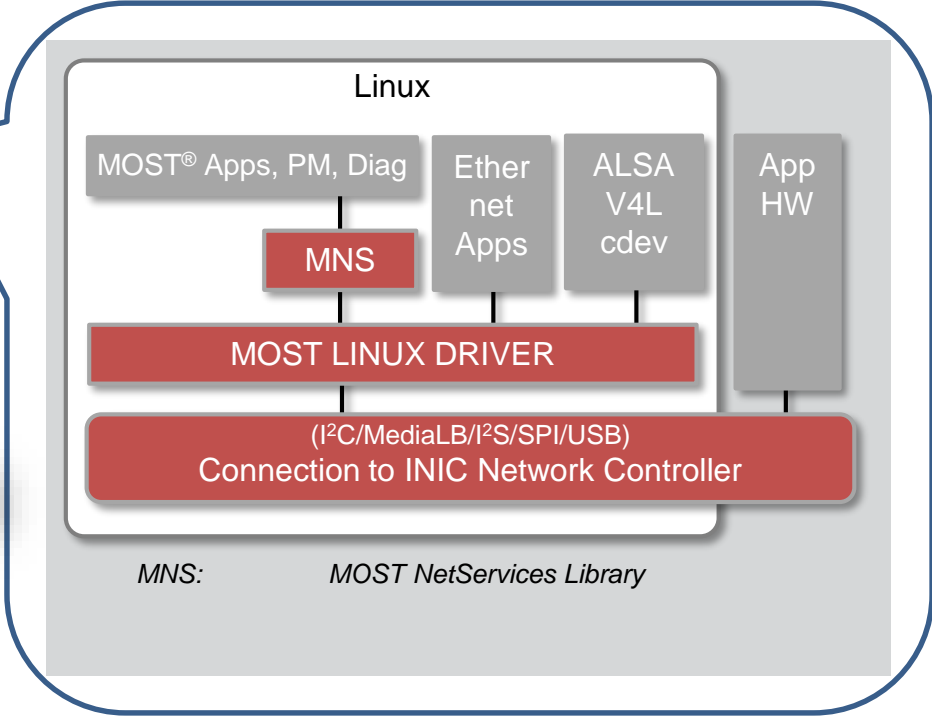
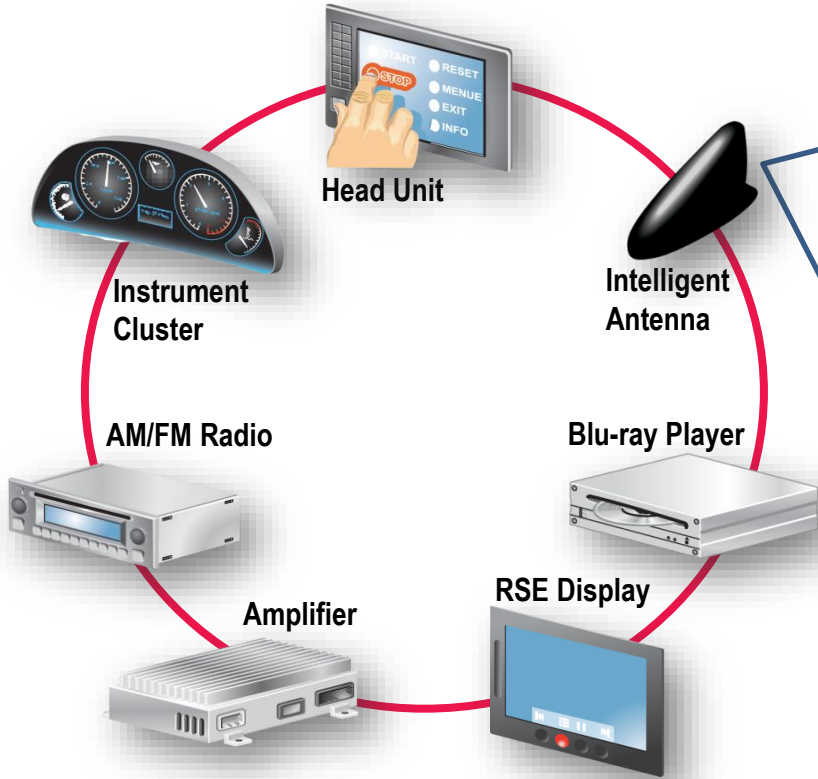
Still the same!

**Device Control** via Functions

# Master node



# Slave node



# NW functionality/interfaces

---

## System Setup

NetworkMaster  
ConnectionMaster  
MOST App  
FBlocks

PowerMgmt  
Diagnosis

## Device Control

MOST App  
FBlocks

Ethernet Apps

## Streaming

ALSA  
Video4Linux  
cdev

App HW

# Linux/MOST – pros and cons

---

- Existing devices can still be used
- Standard MLD available
- Ethernet, ALSA, Video4Linux can interface directly to MLD
- MOST apps still interfacing to well-known MNS and MSMM API
- Reuse of classic MOST app code
- No benefits of AGL





**MICROCHIP**

**Migration from  
Classic MOST<sup>®</sup>  
to  
AGL based UNICENS**

---

# Motivation

---

- **Take advantage of AGL benefits**
- **Take advantage of UNICENS benefits**
  - Centralized network management
  - Slim nodes without SW possible
- **Build new network without legacy MOST<sup>®</sup> devices**

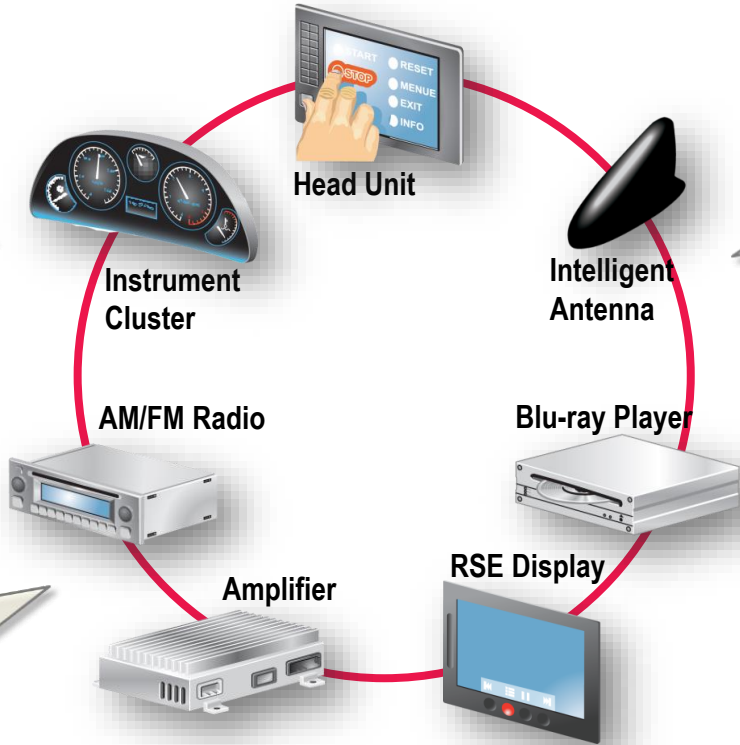
- **Unified Centralized Network Stack (UNICENS) is:**
  - A software stack used for network setup and configuration
  - Supports a synchronous network based on Microchip's INIC network controllers
  - Does not provide a standard for application related communication or system behaviour

# Data types

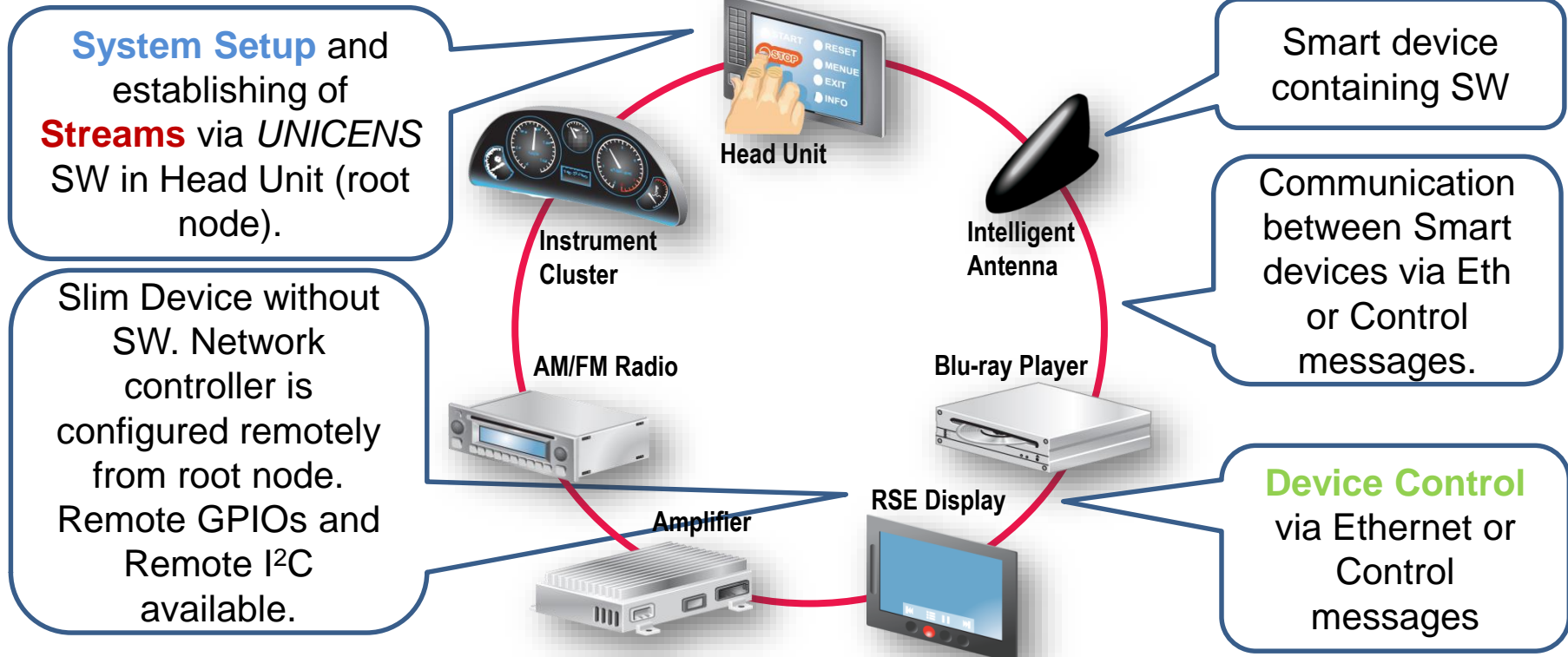
**Control data**  
Control Messages  
used for  
Function Blocks

**Streaming Data**  
Synchronous streams  
(PCM audio etc.) or  
Isochronous Streams  
(MPEG video etc.)

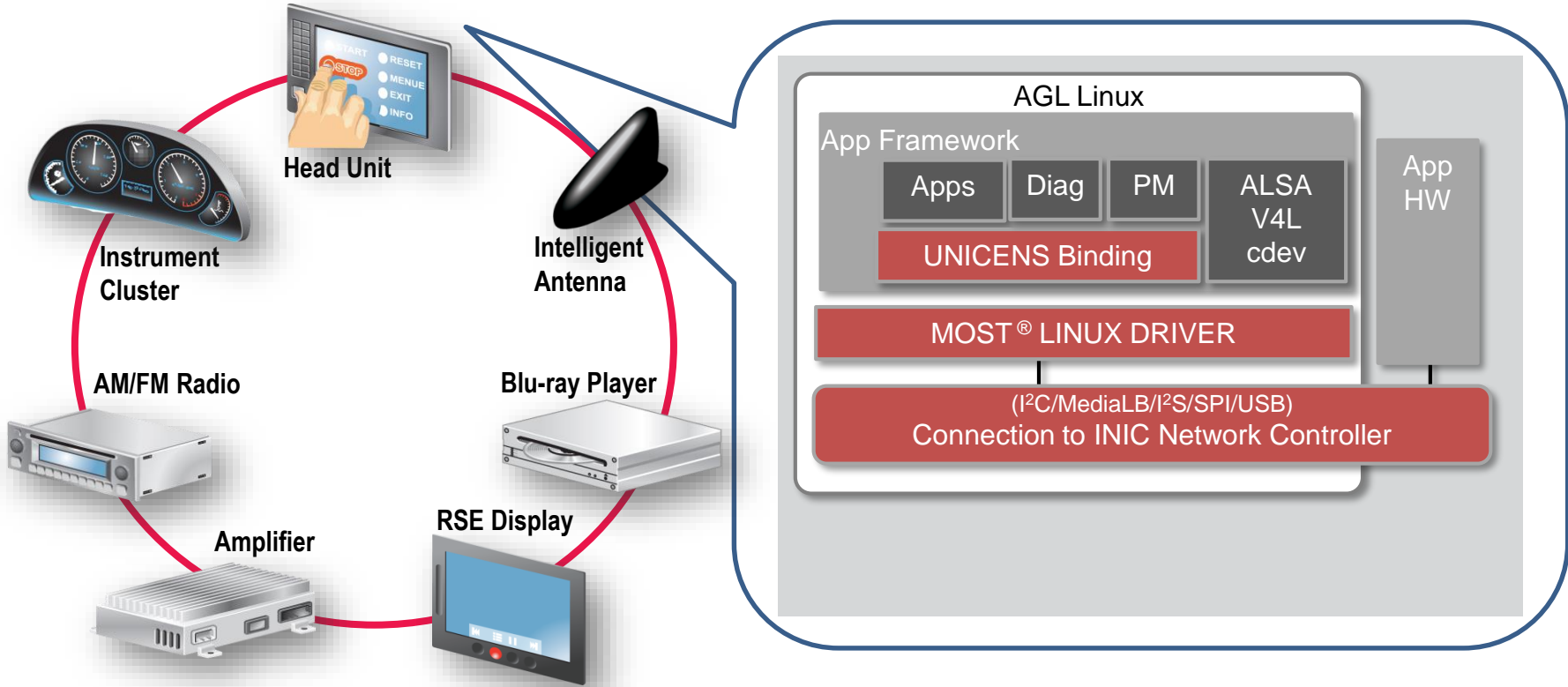
**Packet Data**  
Ethernet packets  
(TCP/IP, UDP, etc.)  
or other packets



# UNICENS network



# Root node



# UNICENS Binding

JSON API

UNICENS  
Integration

Command Queue

Config XML Parser

UNICENS Library

UNICENS Binding

# UNICENS Configuration

- Configuration by a single XML file
- Describes streaming and devices

```
4  <!-- UNICENS Root Node, USB Server Node, -->
5  <Node Address="0x200">
6    <SyncConnection MuteMode="NoMuting">
7      <USBSocket EndpointAddress="0x2" FramesPerTransaction="128"/>
8      <MOSTSocket Route="Music" Bandwidth="4"/>
9    </SyncConnection>
10 </Node>
11
12
13 <!-- 1st Microphone, Slim Node -->
14 <Node Address="0x210">
15   <StreamPort ClockConfig="64Fs" DataAlignment="Left16Bit"/>
16   <SyncConnection MuteMode="NoMuting">
17     <StreamSocket StreamPinID="SRXA0" Bandwidth="4"/>
18     <MOSTSocket Route="Microphone1" Bandwidth="4"/>
19   </SyncConnection>
20 </Node>
```

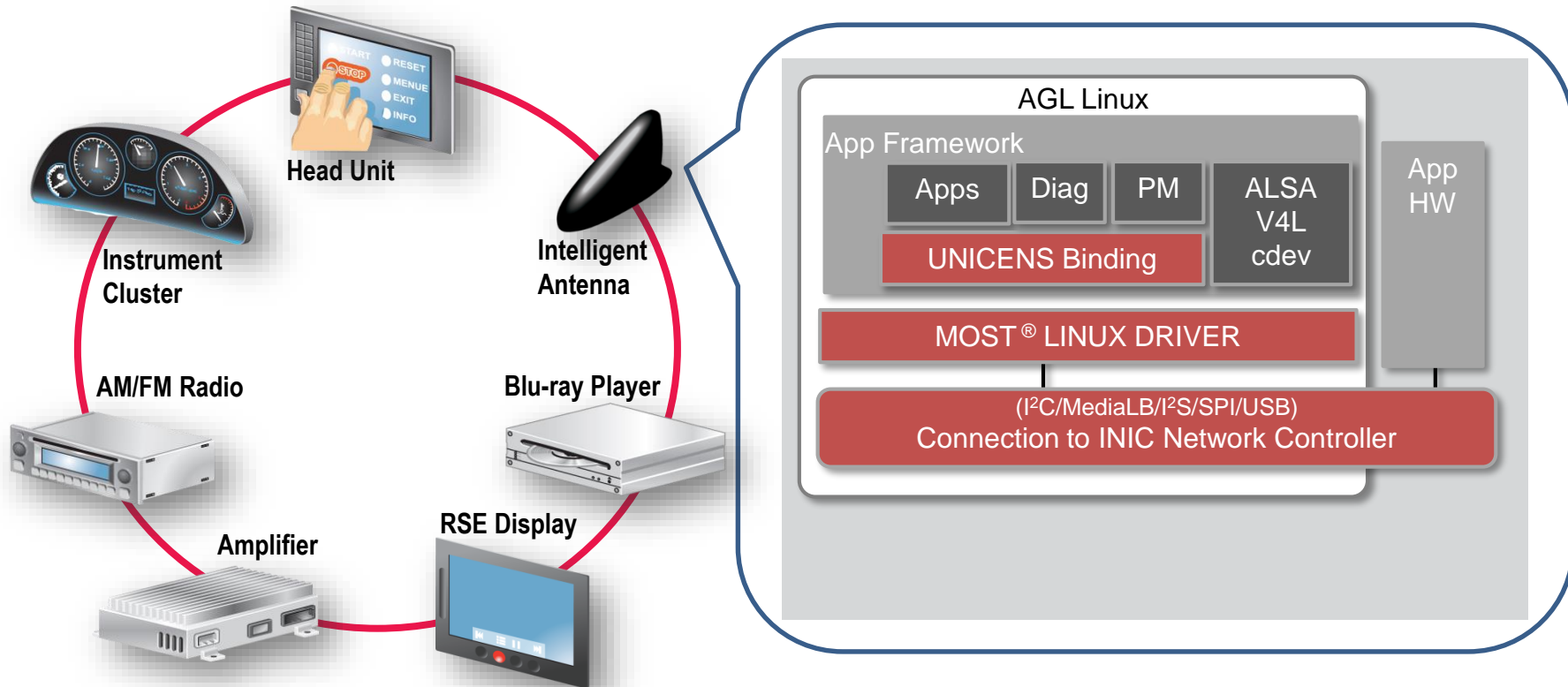


# UNICENS Binding – API

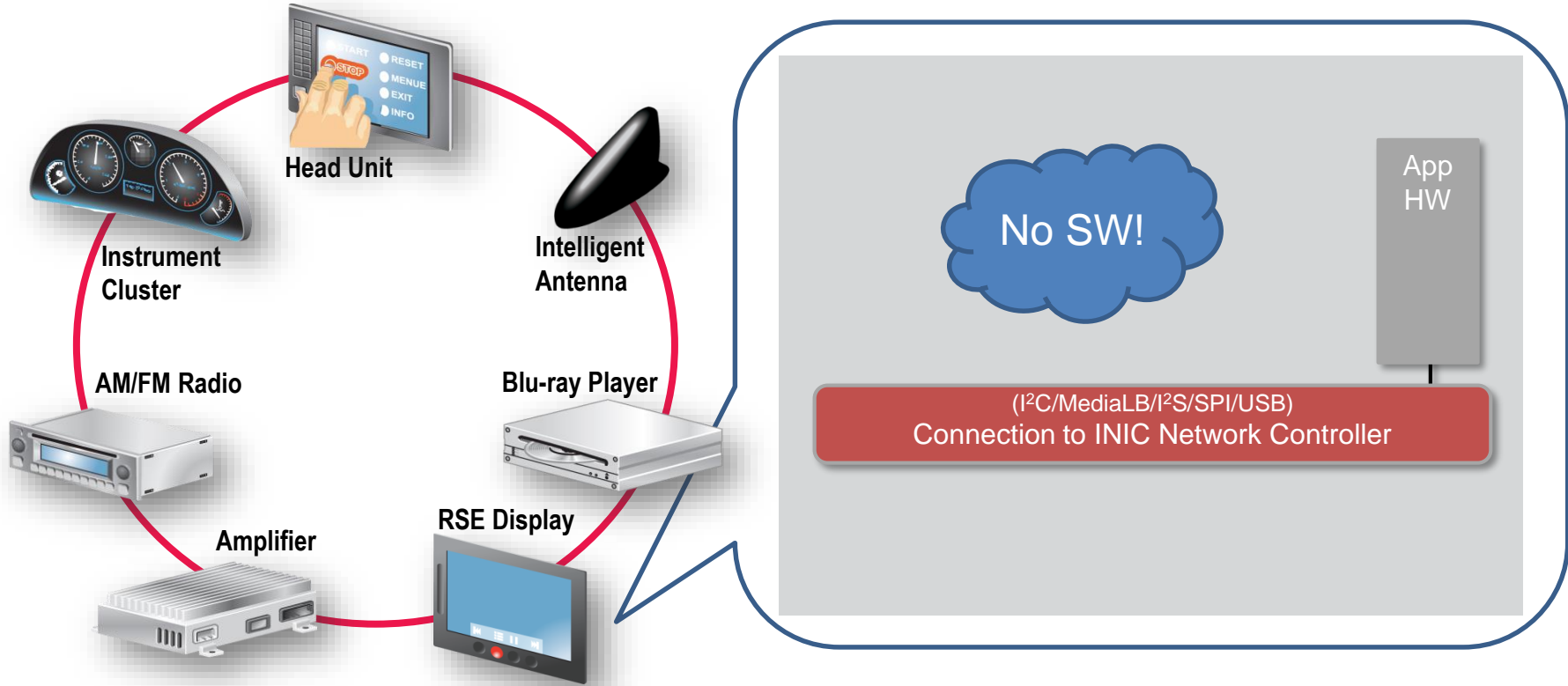
---

- **JSON API using defined verbs**
  - Automatic system setup
  - Notification on network events
  - Tx/Rx of control messages
  - Tx/Rx of remote commands
  - Switching of streaming routes
  - [...]

# Smart node



# Slim node



# NW functionality/interfaces

---

## System Setup

UNICENS  
Binding

PowerMgmt  
Diagnosis

## Device Control

Control Apps

Ethernet Apps

## Streaming

ALSA  
Video4Linux  
cdev

App HW

# AGL/UNICENS – pros/cons

---

- Centralized network management
- Easy network configuration (XML)
- Slim nodes without SW possible
- UNICENS Binding seamlessly integrates in AGL security architecture
- JSON interface to applications
- No legacy MOST devices supported



**MICROCHIP**

**Migration from  
Classic MOST<sup>®</sup>  
to  
AGL based MOST**

---

# Motivation

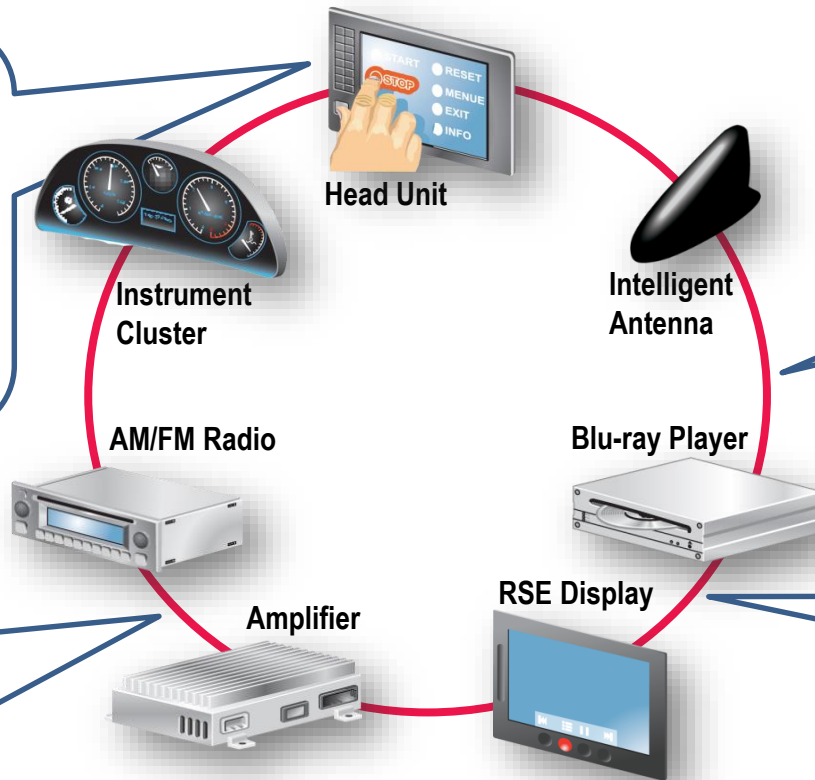
---

- **Replace single devices with AGL devices**
  - Take advantage of AGL benefits
  - No change in network (MOST<sup>®</sup>) behavior
  - Reuse MOST application code whenever possible
- **Keep some existing devices unchanged**

# MOST<sup>®</sup> network

**System Setup** and establishing of **Streams** via *NetworkMaster* and *ConnectionMaster* functionality (MSMM) in Head Unit and Functions in each device

Each node contains *NetServices* SW stack to communicate with network controller

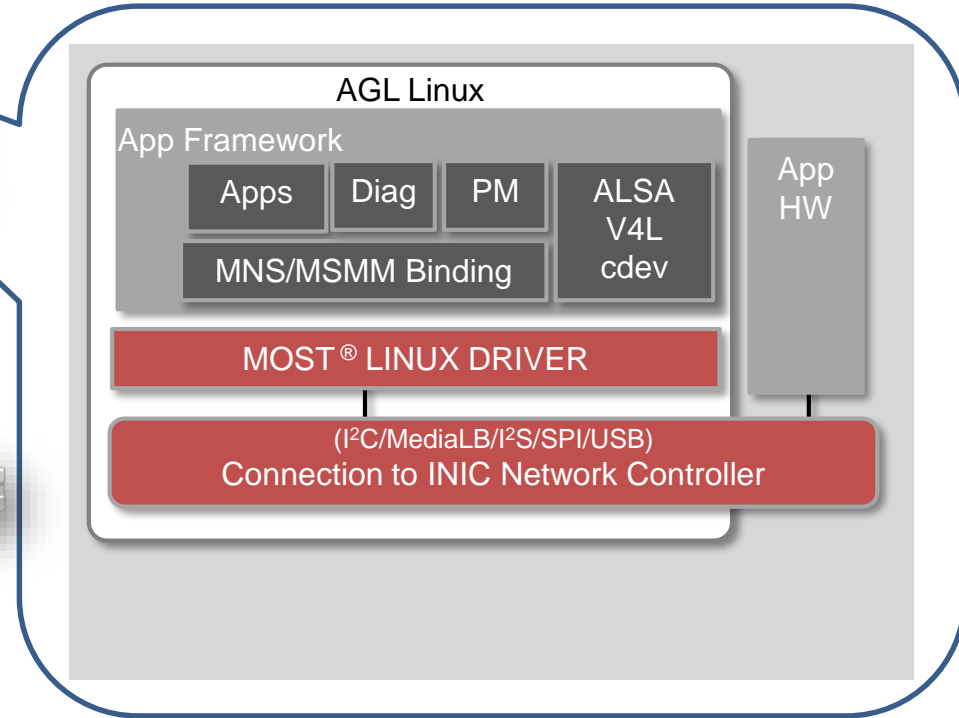
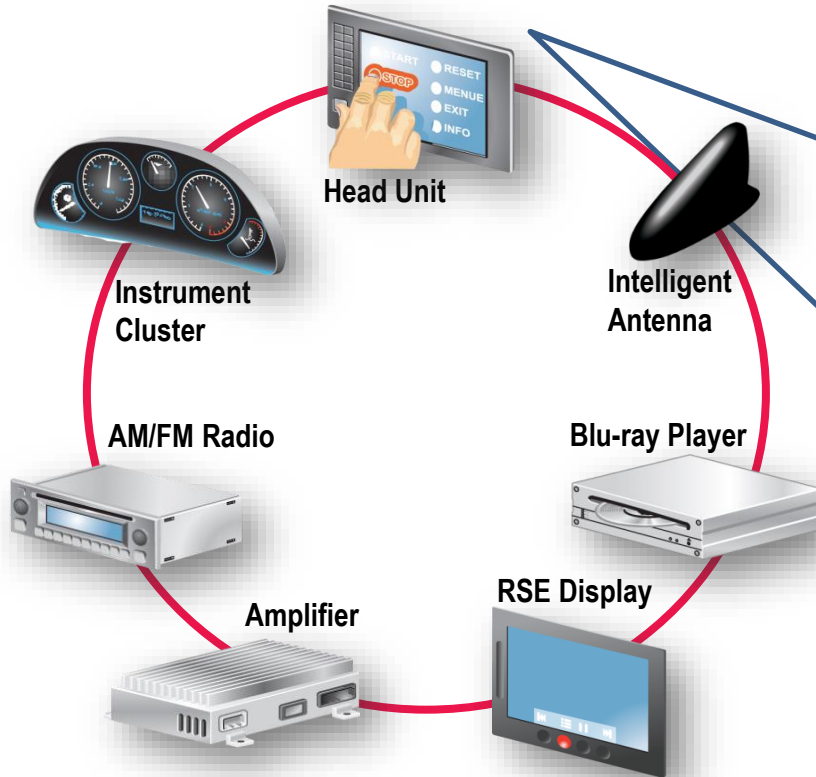


Communication between devices is using *Functions* implemented in each device

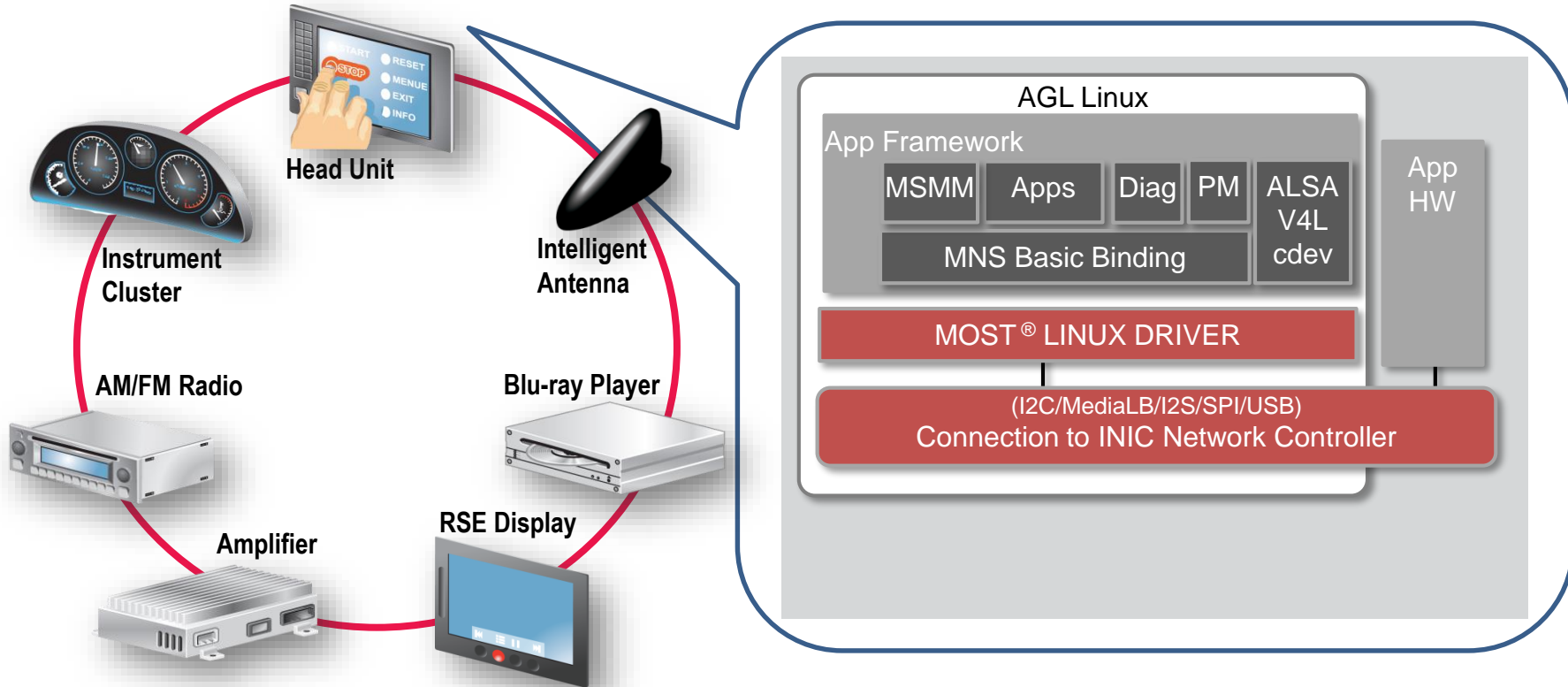
**Device Control** via Functions



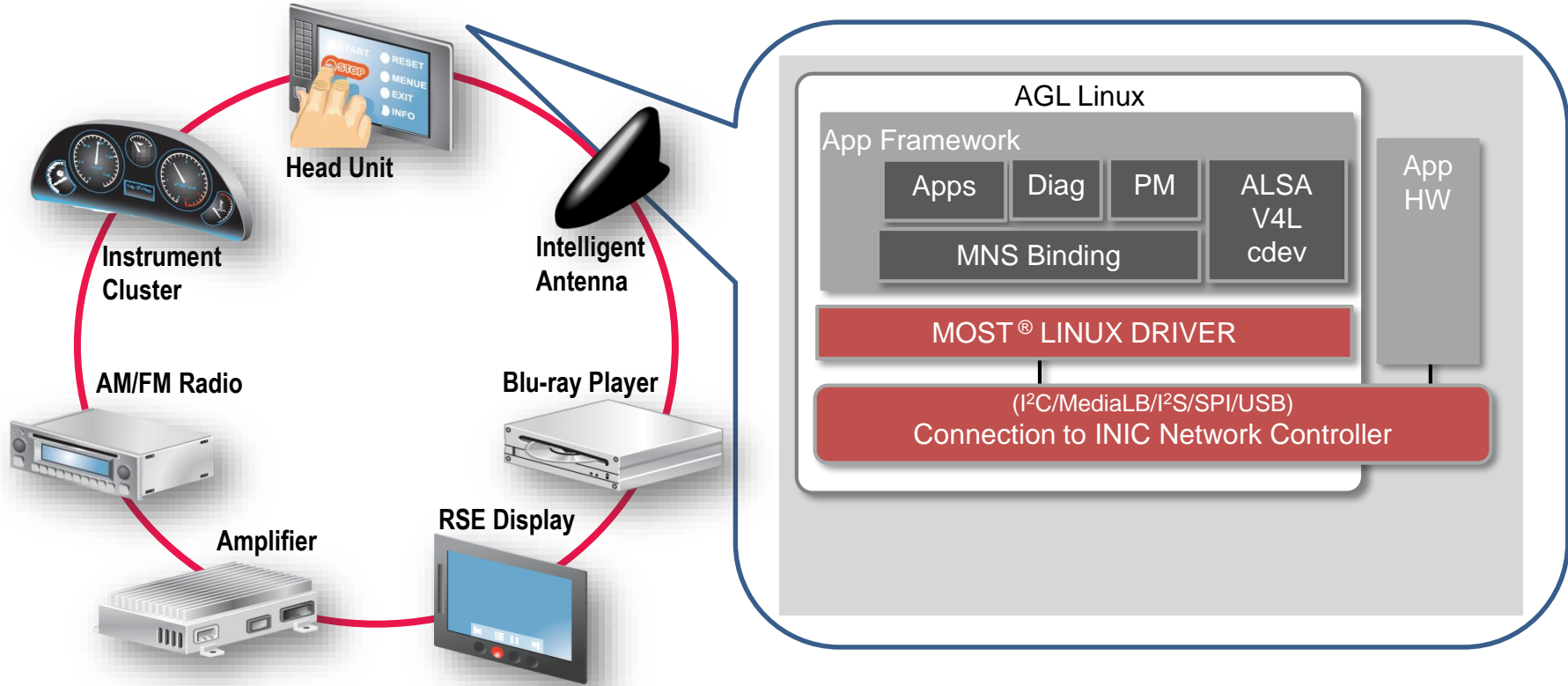
# Master node variant #1



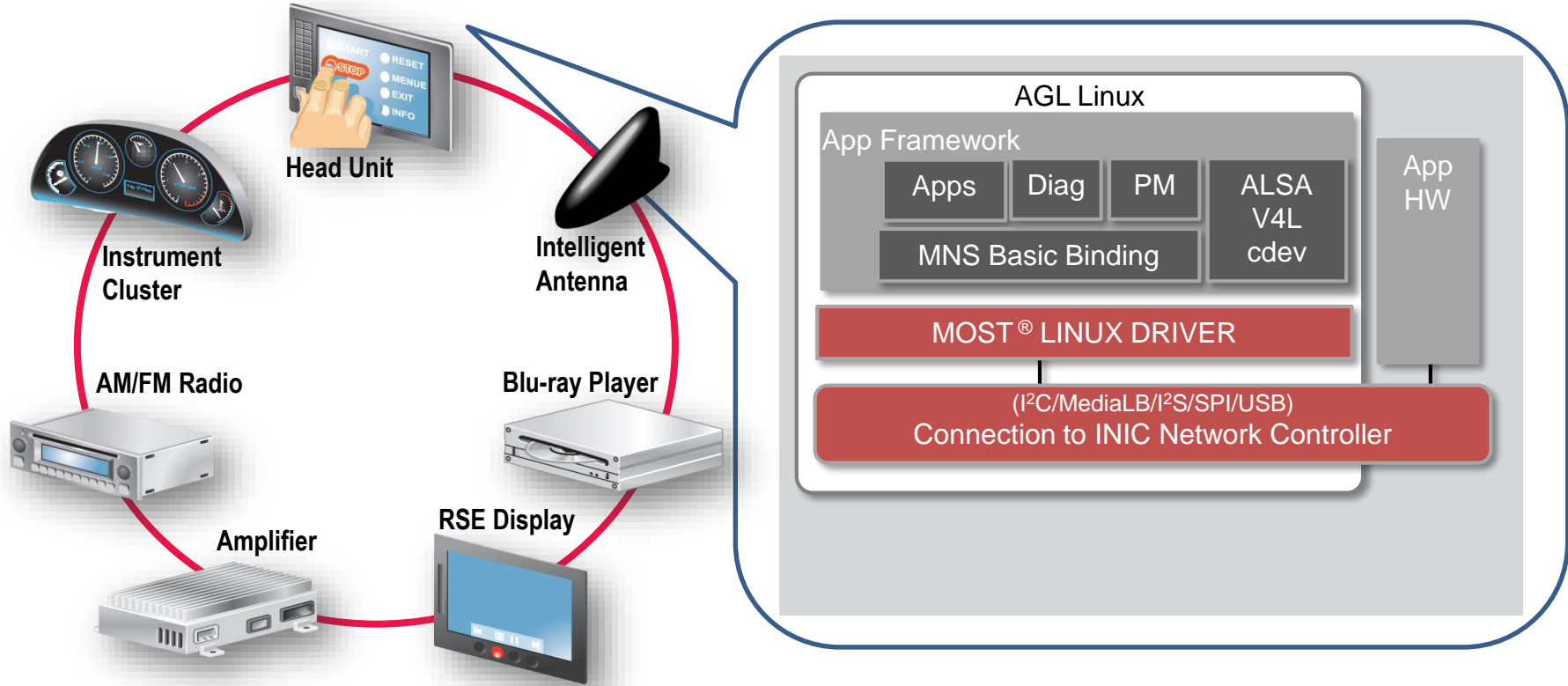
# Master node variant #2



# Slave node variant #1



# Slave node variant #2



# NW functionality/interfaces

---

## System Setup

NetworkMaster  
ConnectionMaster  
MOST App  
FBlocks

PowerMgmt  
Diagnosis

## Device Control

MOST App  
FBlocks

Ethernet Apps

## Streaming

ALSA  
Video4Linux  
cdev

App HW

# AGL/MOST – pros/cons

---

- Existing devices can still be used
- MLD available
- MNS/MSMM, MNS or MNS Basic Binding needs to be developed
- Classic MOST<sup>®</sup> apps need to be adapted to Binding
- Binding might be app-specific



**MICROCHIP**

# Summary



# Summary – Linux/MOST®

---

- **If you go Linux you can:**
  - Take advantage of the MLD
  - Keep MOST as a network
  - Keep single MOST devices unchanged
  - Transfer existing MOST apps to Linux



# Summary – AGL/UNICENS

---

- **If you go AGL with UNICENS you can:**
  - Take advantage of the MLD
  - Implement a new cost effective network
  - Use the available AGL UNICENS binding
  - Easily configure the network
  - Interface applications via Json
  - Easily interface Ethernet applications

# Summary – AGL/MOST®

---

- **If you go AGL with MOST you**
  - Can keep MOST as a network
  - Can keep single MOST devices unchanged
  - Can take advantage of the MLD
  - Need to develop a Binding that interfaces MOST applications to the AGL framework
  - Choose from different Binding approaches



**MICROCHIP**

**Thank you!**





# Trademarks

---

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, ChipKIT, ChipKIT Logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2018, Microchip Technology Incorporated, All Rights Reserved.