Agenda

- Introduction
  - What is HAVi?
  - Advantages
  - Why does the world need HAVi?
- Technology
  - Requirements
  - System Model
    - Control Model
    - Device Classification
      - FAV
      - IAV
      - BAV
      - LAV
    - HAVi Compliance
  - Software Architecture
  - User Interface
    - Level 1
    - Level 2
  - Home Network Configuration
  - Interoperability
    - Level 1
    - Level2
  - IEEE 1394(FireWire)
- Xilinx Value
- Summary

HAVi’s Requirements

- Legacy Device Support
  - The HAVi Architecture supports legacy devices
  - It is important since the transition to networked devices is going to be gradual
  - Characterized by the degree to which they support 1394 and industry standard protocols for 1394 such as IEC 61883
    - Non-1394 devices(most existing CE devices)
    - 1394 devices not supporting the HAVi Architecture

HAVi's Requirements

- Future-Proof Support
  - Great concern of the CE industry since new products should work with existing products
  - The HAVi Architecture supports future devices and protocols
  - Done through several software-based mechanisms which includes:
    - Persistent device-resident information describing capabilities of devices
    - A write-once, run-everywhere language (Java) used for software extensions
    - A device independent representation of user interface elements

HAVi's Requirements

- Each HAVi-compliant device may contain persistent data(Java Byte code) concerning its user interface and device control capabilities
- As manufacturers introduce new models with new features they can modify the byte code shipped with the device
  - Similarly new user interface elements can be added to the stored UI representation on the device.

HAVi's Requirements

- Plug-And-Play Support
  - In the HAVi Architecture, a device configures itself, and integrates itself into the home network, without user intervention
  - Low-level communication services provide notification when a new device is identified on the network
- Flexibility
  - The HAVi Architecture allows devices to present multiple user interfaces, adapting to both the user's needs and the manufacturer's need for brand differentiation

Agenda

- Introduction
  - What is HAVi?
  - Advantages
  - Why does the world need HAVi?
- Technology
  - Requirements
  - System Model
    - Control Model
    - Device Classification
      - FAV
      - IAV
      - BAV
      - LAV
    - HAVi Compliance
  - Software Architecture
  - User Interface
    - Level 1
    - Level 2
  - Home Network Configuration
  - Interoperability
    - Level 1
    - Level2
  - IEEE 1394(FireWire)
- Xilinx Value
- Summary

Control Model

- Devices may exchange control information and data in a peer-to-peer fashion
  - This ensures that, at the communication level, no one device is required to act as a master or controller for the system
- A controller is said to host a “Device Control Module” (DCM) for the controlled device
  - The control interface is exposed via the API of this DCM
- DCMs are a central concept to the HAVi architecture and the source of flexibility in accommodating new devices and features

Control Model

- DCMs can be distinguished as:
- Embedded DCM – A DCM that is part of the resident software on a controller
- Uploaded DCM – A DCM that is obtained from some source external to the controller and dynamically added to the software on the controller
- Native DCM – A DCM that is implemented for a specific platform, it may include machine code for a specific processor or access platform specific APIs

Control Model

- Bytecode DCM – A DCM that is implemented in Java bytecode
- Standard DCM – A DCM that provides the standard HAVI APIs
  - Such a DCM provides basic functionality but is able to control a wide range of devices

Agenda

- Introduction
  - What is HAVi?
  - Advantages
  - Why does the world need HAVi?
- Technology
  - Requirements
  - System Model
    - Control Model
    - Device Classification
      - FAV
      - IAV
      - BAV
      - LAV
    - HAVi Compliance
  - Software Architecture
  - User Interface
    - Level 1
    - Level 2
  - Home Network Configuration
  - Interoperability
    - Level 1
    - Level2
  - IEEE 1394(FireWire)
- Xilinx Value
- Summary

HAVi Device Classes

Controller Devices

Full AV device (FAV)
- Download and execute all HAVi applications
- Download and execute DCM

Intermediate AV device (IAV)
- Ability to communicate with other HAVi device
- Ability to execute limited applications
- Offers own control service
- Ability to host other known device

Base AV device (BAV)
- Offers own information in ROM

Legacy AV device (LAV)
- Conventional devices with NO HAVi SDD data (ROM)

Full AV Device

- Has a rich set of resources and is capable of supporting a complex software environment
- The primary distinguishing feature of an FAV is the presence of a runtime environment for Java bytecode
  - This allows an FAV to upload bytecode from other devices
  - It provides enhanced capabilities for their control
- Likely candidates for FAV devices:
  - STB
  - DTV,
  - General purpose home control devices
  - Home PC's

Intermediate AV Device

- IAV devices are generally lower in cost than FAV devices and more limited in resources
- They do not provide a runtime environment for Java bytecode
  - They cannot act as controllers for arbitrary devices within the home network
- IAV may provide native support for control of particular devices on the home network.

Base AV Device

- Devices that choose to implement future-proof behavior by providing uploadable Java bytecode
  - But do not host any of the software elements of the HAVi Architecture
- These devices can be controlled by an FAV device via the uploadable bytecode or from an IAV device via native code
- Communication between a FAV or IAV device and a BAV device requires that HAVi commands be translated to and from the command protocol used by the BAV device

Legacy AV Device

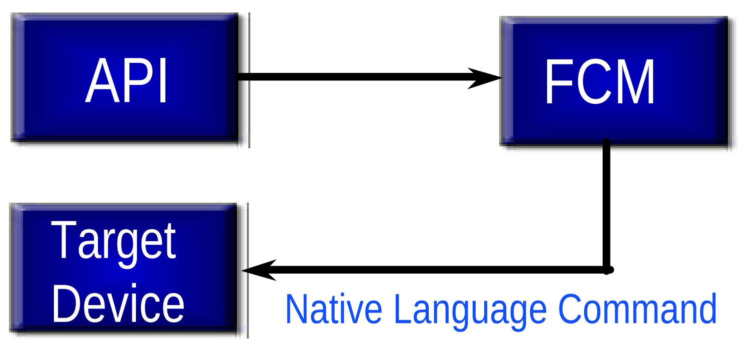
- LAV devices are not aware of the HAVi Architecture
- These devices use proprietary protocols for their control, and quite frequently have simple control-only protocols
- Such devices can work in the home network but require that FAV or IAV devices act as a gateway
- Communication between a FAV or IAV device and legacy device requires that HAVi commands be translated to and from the legacy command protocol

HAVi Compliance

- Each HAVi compliant device (FAV, IAV and BAV) shall:
  - Support one or more of: 1394-1995, 1394-a or 1394-b
  - Provide HAVi SDD data in a IEEE 1212 configuration ROM
  - If the device sources or sinks a stream type for which IEC 61883 transmission has been specified, then the device should support:
    - The PCR and CMP rules for isochronous connections as defined in IEC 61883.1
    - The CIP protocol as defined in IEC 61883.1
    - The CIP format specific definition in the corresponding part of IEC 61883

HAVi Functional Component Module (FCM)

- Tuner
- VCR
- Clock
- Camera
- AV Disc
- Amplifier
- Display
- AV Display
- Modem
- Web Proxy



Agenda

- Introduction
  - What is HAVi?
  - Advantages
  - Why does the world need HAVi?
- Technology
  - Requirements
  - System Model
    - Control Model
    - Device Classification
      - FAV
      - IAV
      - BAV
      - LAV
    - HAVi Compliance
  - Software Architecture
  - User Interface
    - Level 1
    - Level 2
  - Home Network Configuration
  - Interoperability
    - Level 1
    - Level2
  - IEEE 1394(FireWire)
- Xilinx Value
- Summary

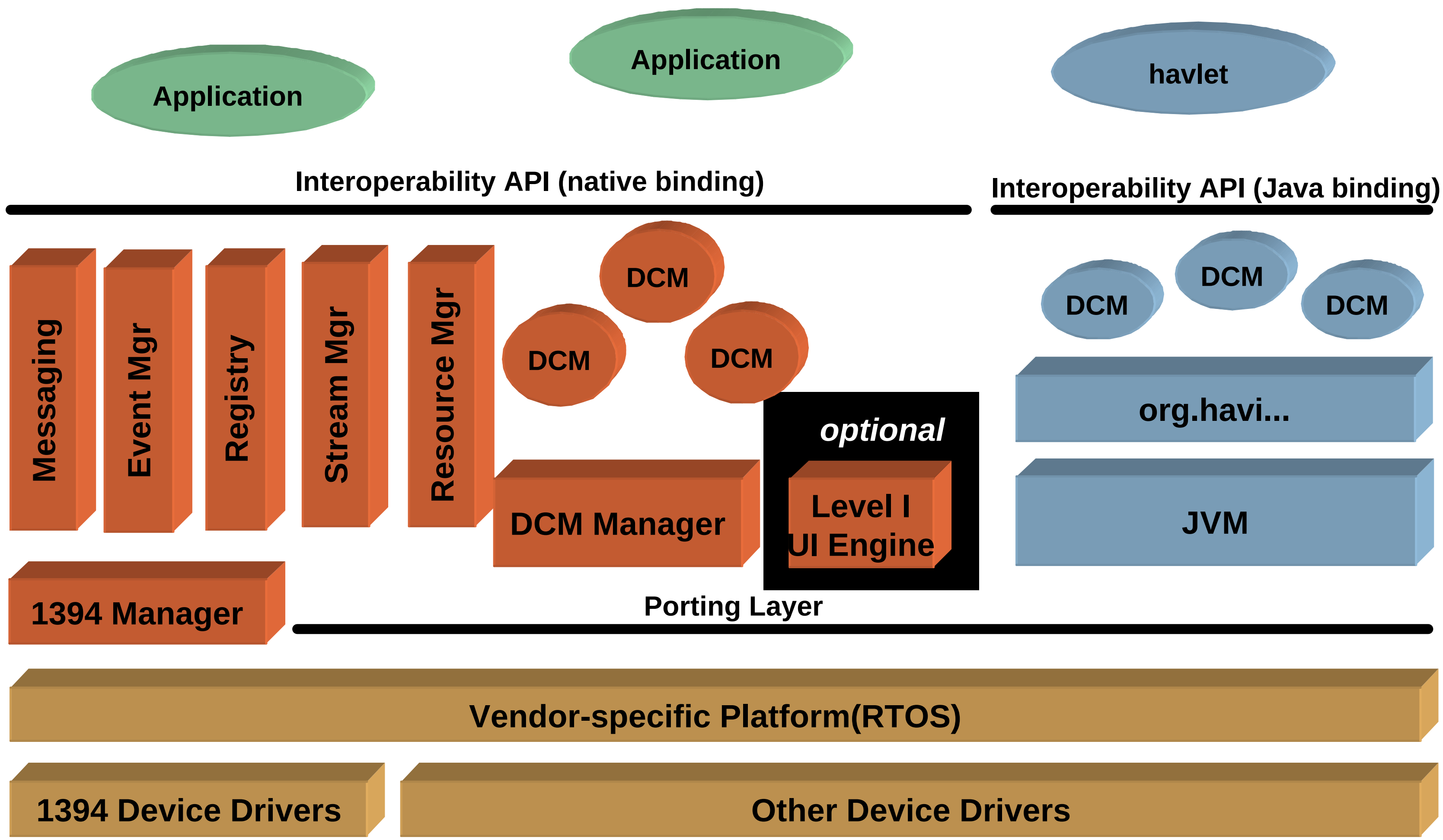
HAVi Architecture

- A set of software elements along with the protocols and APIs needed to achieve interoperability
- Device abstraction and device control models
- An addressing scheme and lookup service for devices and their resources
- An open execution environment supporting visual presentation and control of devices, and providing runtime support for third party applications

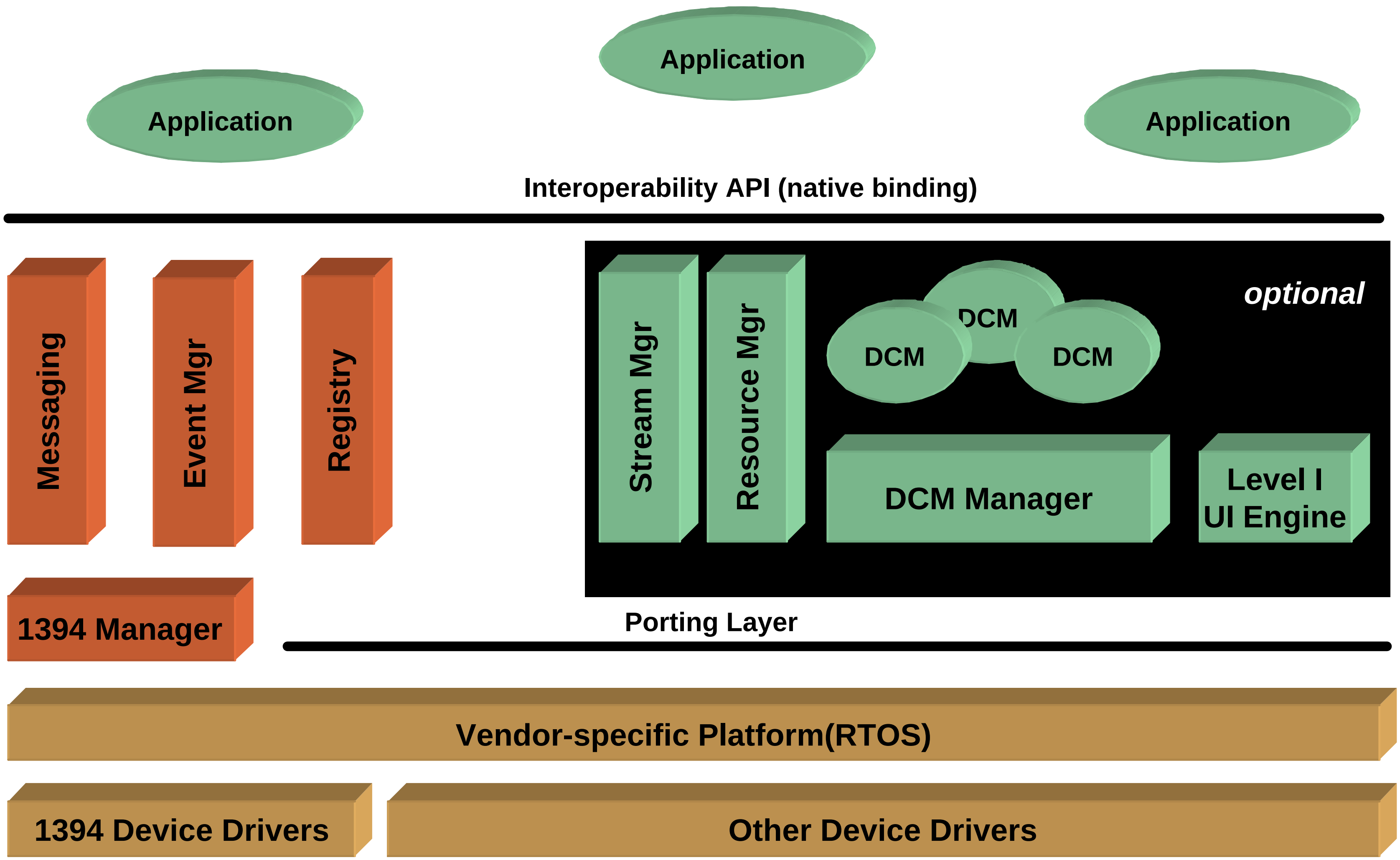
HAVi Architecture

- Communication mechanisms for extending the environment dynamically through plug-and- play capabilities
- The HAVi architecture is:
  - Open
  - Scaleable in implementation complexity
  - Platform-independent and language neutral
    - HAVi can be implemented in any programming language and on any CPU or real-time operating system

HAVi Architecture (FAV)



HAVi Architecture (IAV)



HAVi Architecture

- 1394 Communication Media Manager
  - Allows other software elements to perform asynchronous and isochronous communication over 1394
- Messaging System
  - Responsible for passing messages between software elements
- Registry
  - Serves as a directory service, allows any object to locate another object on the home network

HAVi Architecture

- Event Manager
  - Serves as an event delivery service
    - An event is the change in state of an object or of the home network
- Stream Manager
  - Responsible for managing real-time transfer of AV and other media between functional components
- Resource Manager
  - Facilitates sharing of resources and scheduling of actions

HAVi Architecture

- Device Control Module (DCM)
  - A software element used to control a device
  - DCMs are obtained from DCM code units
  - Within a DCM code unit are:
    - Code for the DCM itself
    - Code for Functional Component Modules (FCMs) for each functional component within the device
- DCM Manager
  - Responsible for installing and removing DCM code units on FAV and IAV devices

Content of Home Network HAVi Device

- Application Module
  - Is a software element that may provide a DDI(Data Driven Interaction) interface and/or a havlet
  - Havlet is a HAVi Java application that is uploaded on the request of a controller from a DCM or application module
- Self Describing Device (SDD) data
  - Contains descriptive information about the device and its capabilities
  - Follows the IEEE 1212 addressing scheme used for Configuration ROM
  - May include a DCM code unit and vendor-specific data for constructing user interface elements

Content of Home Network HAVi Device

- Java Runtime Environment
  - Provides an execution environment for uploaded DCMs and applications implemented using Java bytecode
- DDI Controller
  - A software element involved with user interaction
  - The DDI (Data Driven Interaction) Controller handles user input and interprets (renders) DDI elements

Agenda

- Introduction
  - What is HAVi?
  - Advantages
  - Why does the world need HAVi?
- Technology
  - Requirements
  - System Model
    - Control Model
    - Device Classification
      - FAV
      - IAV
      - BAV
      - LAV
    - HAVi Compliance
  - Software Architecture
  - User Interface
    - Level 1
    - Level 2
  - Home Network Configuration
  - Interoperability
    - Level 1
    - Level2
  - IEEE 1394(FireWire)
- Xilinx Value
- Summary

User Interface Support

- Level 1 UI
  - Intended for IAVs and is called Data Driven Interaction(DDI)
  - DDI elements can be loaded from a DDI Target(typically DCM) and displayed by a DDI controller
- Level 2 UI
- Constructed by bytecode applications running on FAVs
  - Support for different pixel aspect ratios, screen aspect ratios and screen sizes
  - Support for alpha blending and video / image layering
  - Support for remote control input

Agenda

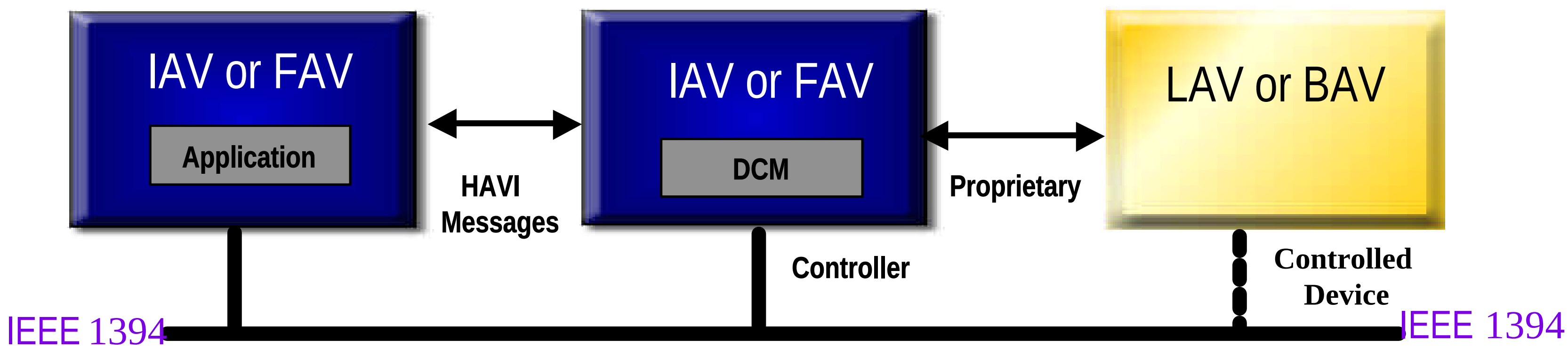
- Introduction
  - What is HAVi?
  - Advantages
  - Why does the world need HAVi?
- Technology
  - Requirements
  - System Model
    - Control Model
    - Device Classification
      - FAV
      - IAV
      - BAV
      - LAV
    - HAVi Compliance
  - Software Architecture
  - User Interface
    - Level 1
    - Level 2
  - Home Network Configuration
  - Interoperability
    - Level 1
    - Level2
  - IEEE 1394(FireWire)
- Xilinx Value
- Summary

Home Network Configurations

- It defines APIs and messaging protocols so that interoperability is assured
- It defines how future devices and services can be integrated into the architecture
- The HAVi Architecture makes no restrictions on what types of devices must be present in the home network
  - Networks without FAV devices
  - Networks with multiple FAV devices
  - Networks with LAV and BAV devices only

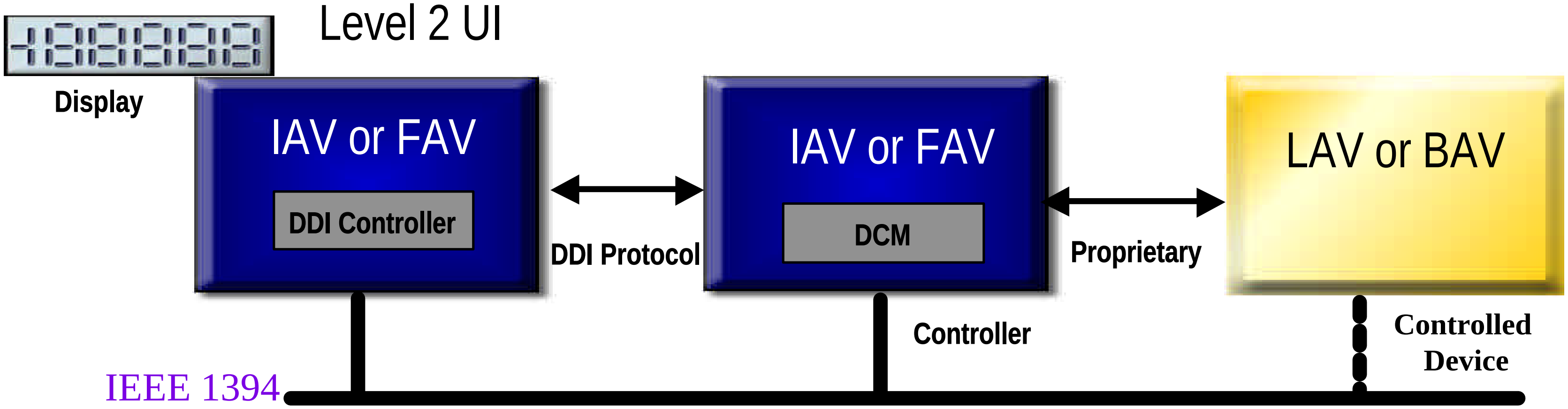
Home Network Configurations

- IAV or FAV as Controller
  - IAV and FAV devices act as controllers for the other device classes and provide a platform for the system services comprising the HAVi Architecture
  - FAVs may host Java bytecode DCMs
  - The primary role of a controller is to provide a runtime environment for DCMs



Home Network Configurations

- IAV or FAV as display
  - IAVs and FAVs will have an associated display device that is used for display of AV content and GUIs
  - Devices without display will cooperate with other IAV or FAV devices with display capability
    - A display capable IAV is required to support a DDI Controller
    - A display capable FAV is required to support a DDI Controller and a Level 2 UI



Home Network Configurations

- Peer-to-peer Architecture between FAVs & IAVs
  - If there are more than one FAV or IAV, each controller cooperates with other controllers to ensure that services are provided to the user
  - An example: a device without display capabilities uses a remote device to display DCM user interfaces
- IAVs as Controller and Display
  - Embedded DCMs can be implemented as native applications on the IAV device and can use native interfaces to access the IAV's display and other resources

Agenda

- Introduction
  - What is HAVi?
  - Advantages
  - Why does the world need HAVi?
- Technology
  - Requirements
  - System Model
    - Control Model
    - Device Classification
      - FAV
      - IAV
      - BAV
      - LAV
    - HAVi Compliance
  - Software Architecture
  - User Interface
    - Level 1
    - Level 2
  - Home Network Configuration
  - Interoperability
    - Level 1
    - Level2
  - IEEE 1394(FireWire)
- Xilinx Value
- Summary

Interoperability

- The first and foremost goal of the HAVi Architecture is to support interoperability between AV equipment
- Level 1 Interoperability
  - Defines and uses a generic set of control messages (commands) that enable one device to talk to another device
  - Defines and uses a set of event messages that it should reasonably expect from the device

Interoperability (Level 1)

- Following mechanisms are required:
- Device Discovery
  - Each device in the home network needs a well-defined method that allows it to advertise its capabilities to others
  - SDD data contains information about the device which can be accessed by other devices
  - The SDD data contains, as a minimum, enough information to allow instantiation of an embedded DCM
    - This results in registration of device capabilities with the HAVI Registry

Interoperability (Level 1)

- Communication
  - A general communication facility is needed to access the capabilities of another device on the network
  - This service is provided by the HAVI Messaging Systems and DCMs
  - The application sends HAVI messages to DCMs, the DCM then engages in proprietary communication with the device

Interoperability (Level 1)

- HAVI Message Set
  - A well defined set of messages that must be supported by all devices of a particular class
  - This ensures that a device can work with existing, as well as future devices, irrespective of the manufacturer
  - The HAVI message set includes those messages used for the DDI protocol and so allows DCMs (and applications) to construct a UI on display-capable IAVs and FAVs

Interoperability(Level 2)

- Level 2 Interoperability
  - Allows a device to communicate to other devices any additional functionality not present in embedded DCMs
  - The HAVi Architecture allows uploaded DCMs as an alternative to embedded DCMs
    - To support non-standard features of existing products
    - To support future products
  - Level 2 only requires that one device provide a runtime environment for the uploaded DCM obtained from the new device

Agenda

- Introduction
  - What is HAVi?
  - Advantages
  - Why does the world need HAVi?
- Technology
  - Requirements
  - System Model
    - Control Model
    - Device Classification
      - FAV
      - IAV
      - BAV
      - LAV
    - HAVi Compliance
  - Software Architecture
  - User Interface
    - Level 1
    - Level 2
  - Home Network Configuration
  - Interoperability
    - Level 1
    - Level2
  - IEEE 1394(FireWire)
- Xilinx Value
- Summary
- Summary

IEEE 1394(FireWire)

- A hardware and software standard for transporting data at 100, 200, 400, or 800 megabits per second (Mbps)
- A digital interface - there is no need to convert digital data into analog and tolerate a loss of data integrity
- Physically small - the thin serial cable can replace larger and more expensive interfaces
- Easy to use - there is no need for terminators, device IDs, or elaborate setup

IEEE 1394 (FireWire)

- Hot pluggable - users can add or remove 1394 devices with the bus active
- Inexpensive - priced for consumer products
- Scaleable architecture - may mix 100, 200, and 400 Mbps devices on a bus
- Flexible topology - support of daisy chaining and branching for true peer-to-peer communication
- Non-proprietary - there is no licensing problem to use for products

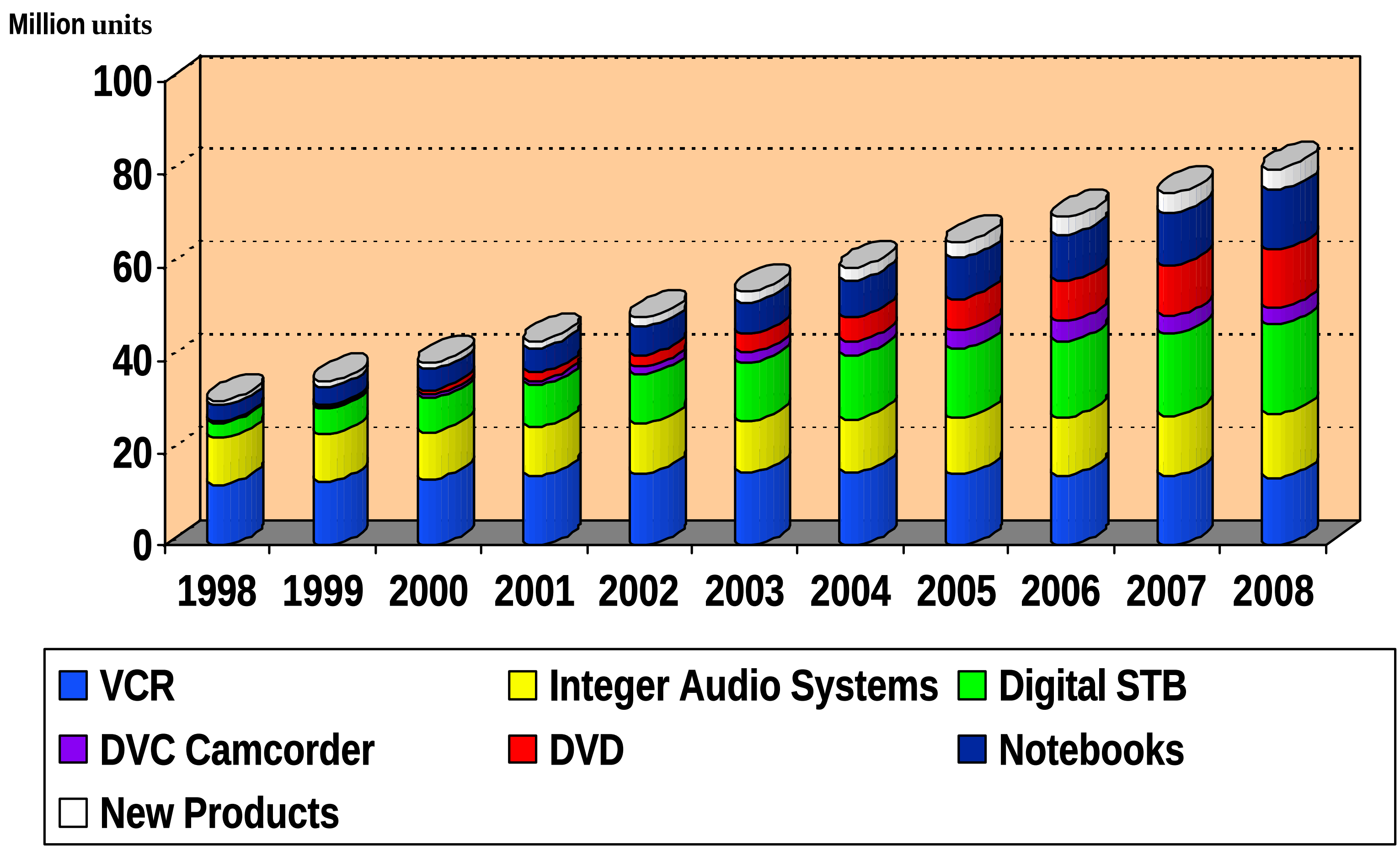
IEEE 1394 Data Transfer

- There are two types of IEEE 1394 data transfer:
- Asynchronous
  - Data is sent in one direction followed by acknowledgment to the requestor.
- Isochronous
  - Data channels provide guaranteed data transport at a pre-determined rate
  - This is especially important for time-critical multimedia data where just-in-time delivery eliminates the need for costly buffering

Multimedia Bandwidth Requirements

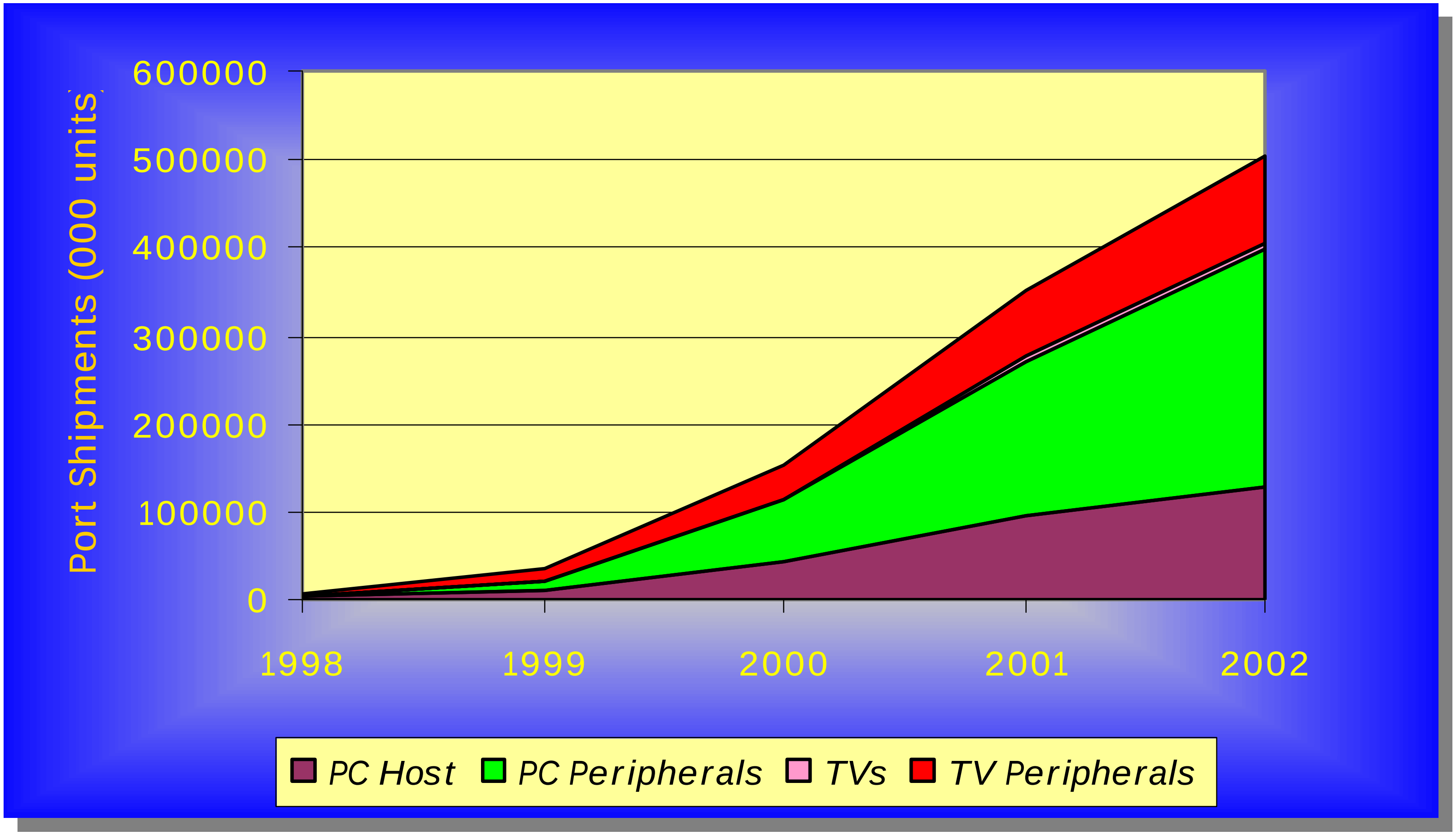
- High Quality Video
  - Digital Data = (30 frames / second) (640 x 480 Pixels) (24-bit color / pixel) = 221 Mbps
- Reduced Quality Video
  - Digital Data = (15 frames / second) (320 x 240 Pixels) (16-bit color / pixels) = 18 Mbps
- High Quality Audio
  - Digital Data = (44,100 audio samples / sec) (16-bit audio samples) (2 audio channels for stereo) = 1.4 Mbps
- Reduced Quality Audio
  - Digital Data = (11,050 audio samples / sec) (8-bit audio samples) (1 audio channel for monaural) = 0.1 Mbps

IEEE 1394 Usage & Growth



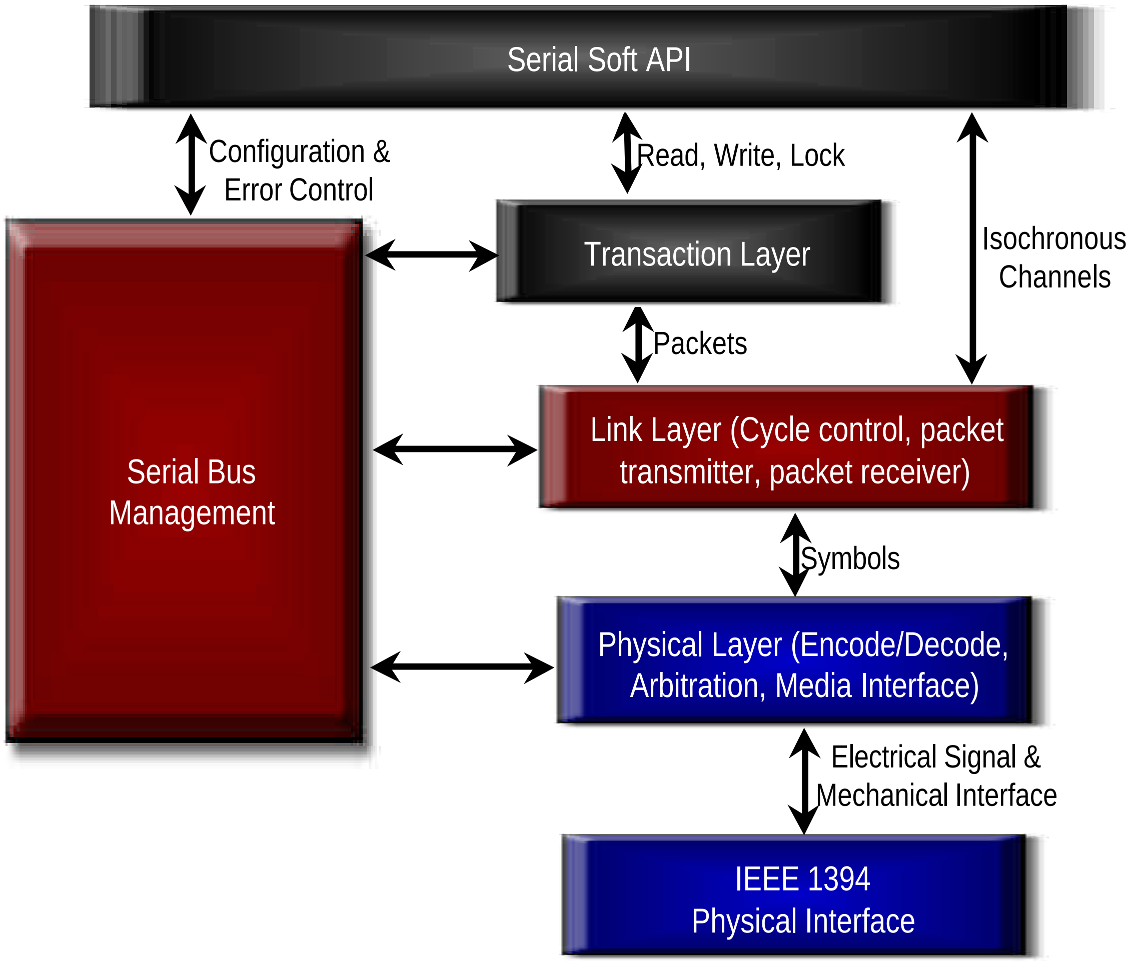
Source: In-stat

1394 Market Forecast



Source - In-Stat

IEEE 1394 Protocol Stack



IEEE 1394 Operation

- To transmit data, a 1394 device first requests control of the physical layer
- With asynchronous transport, the address of both the sender and the receiver is transmitted followed by the actual packet data
- Once the receiver accepts the packet, a packet acknowledgment is returned to the original sender
- To improve throughput, the sender may continue transmission until 64 transactions are outstanding

IEEE 1394 Operation

- With isochronous transport, the sender requests an isochronous channel with a specific bandwidth
- Isochronous channel IDs are transmitted followed by the packet data
- The receiver monitors the incoming data's channel ID and accepts only data with the specified ID