



Distributed Embedded Systems Communications with the CCR, DSS and VPL

Arvindra Sehmi
Director, D&PE
Microsoft Corporation
blogs.msdn.com/asehmi
asehmi@microsoft.com



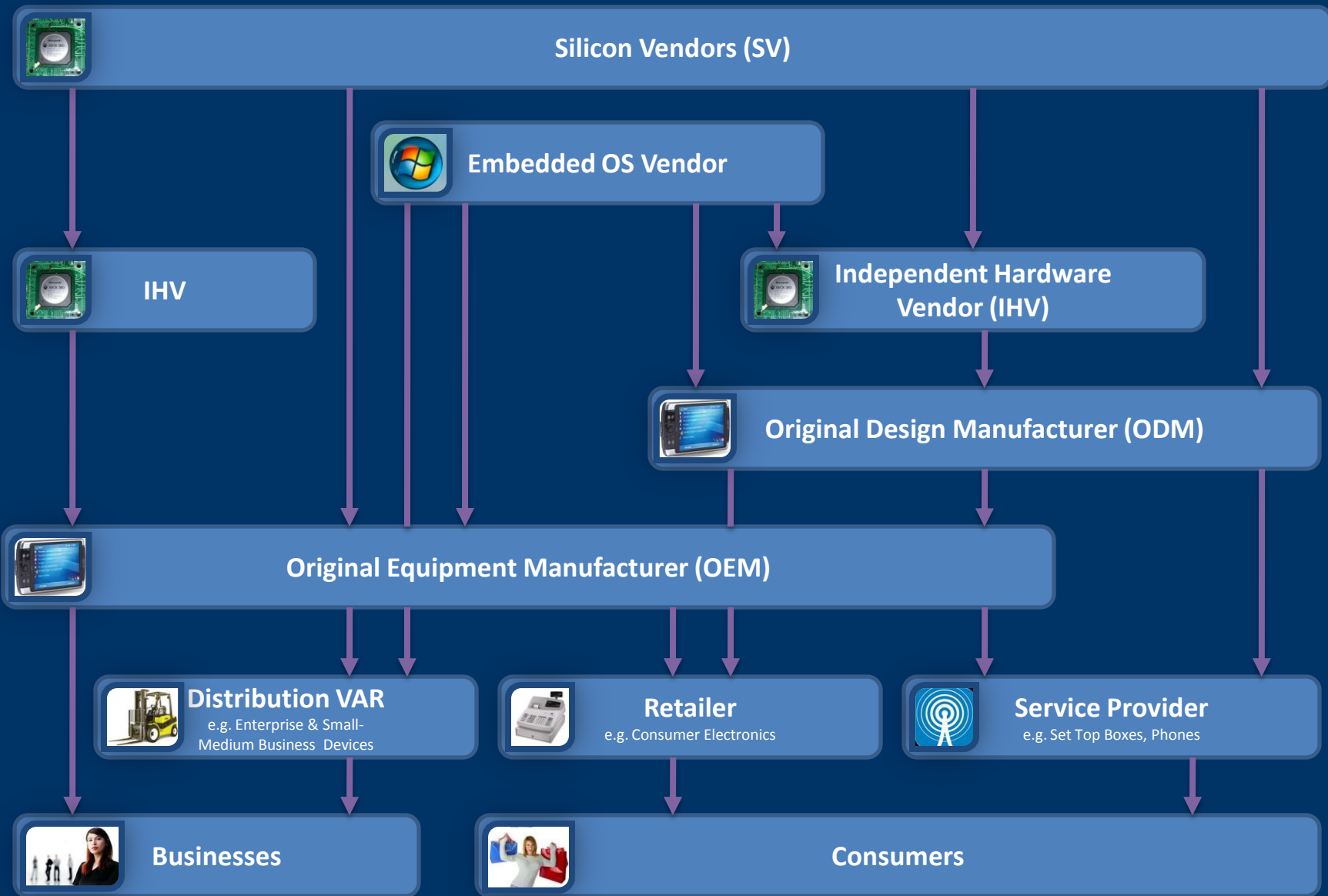
What we'll cover today...

- Embedded Systems Development
- Distributed Embedded Systems Communications
- Some Examples

Who's Who in the Embedded Systems Ecosystem?

The Making of an Embedded Device

Embedded Device Value Chain

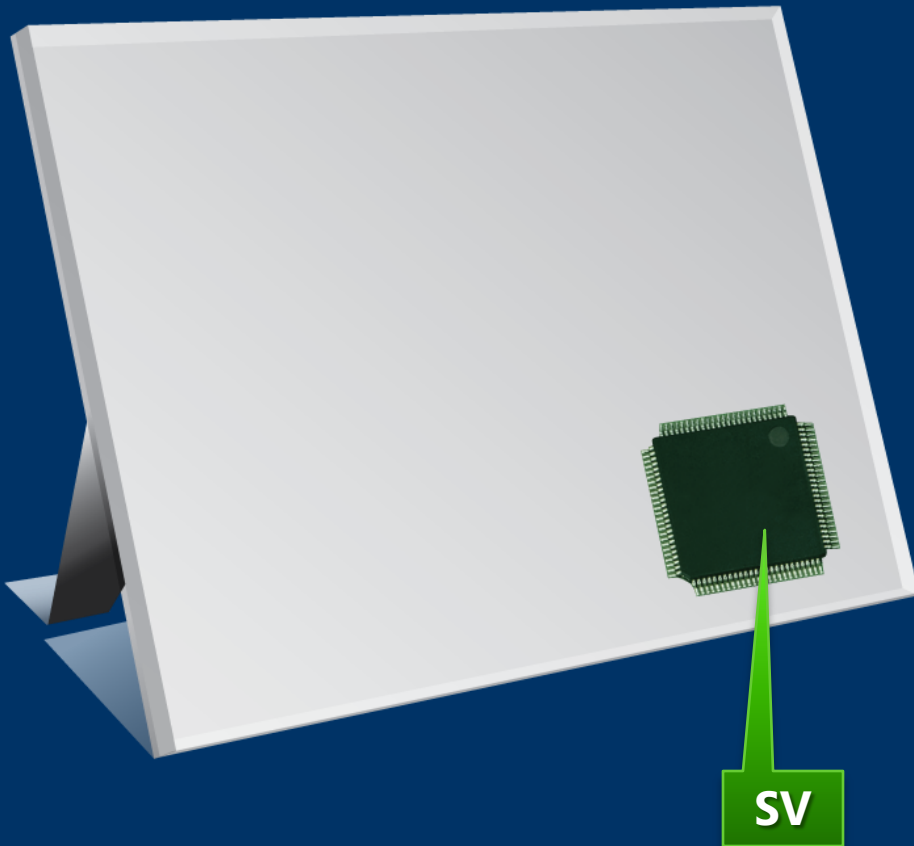


The Making of an Embedded Device



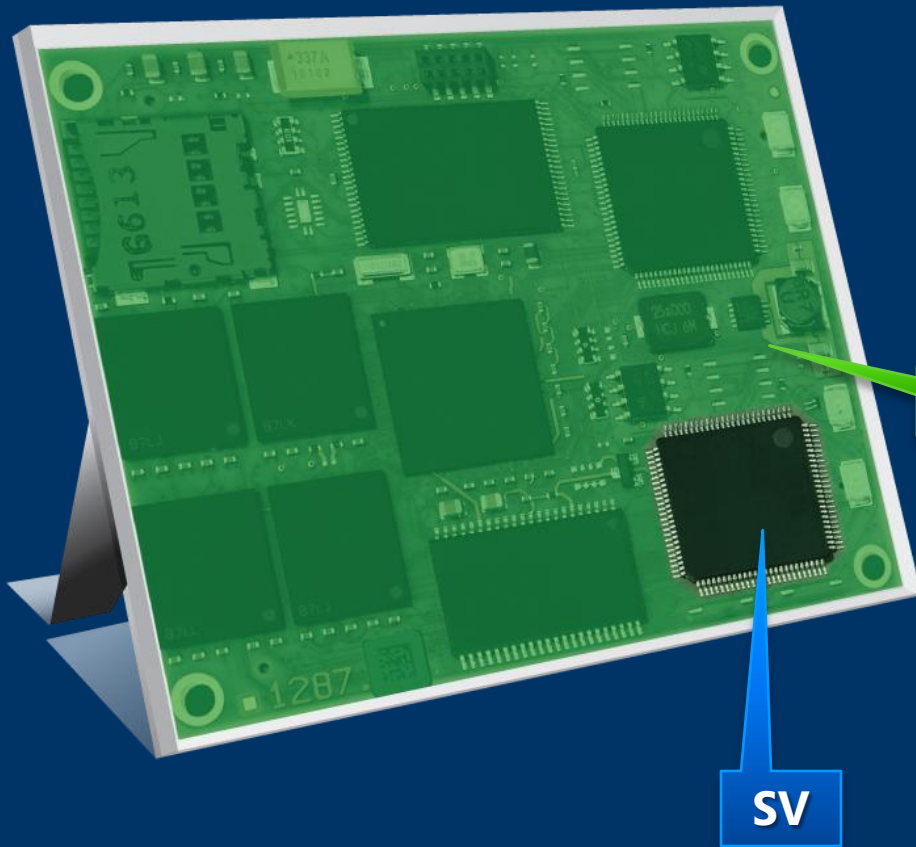
- What does it take to get a device to market?
- Take a peek inside a Digital Picture Frame and find out...

Silicon Vendor



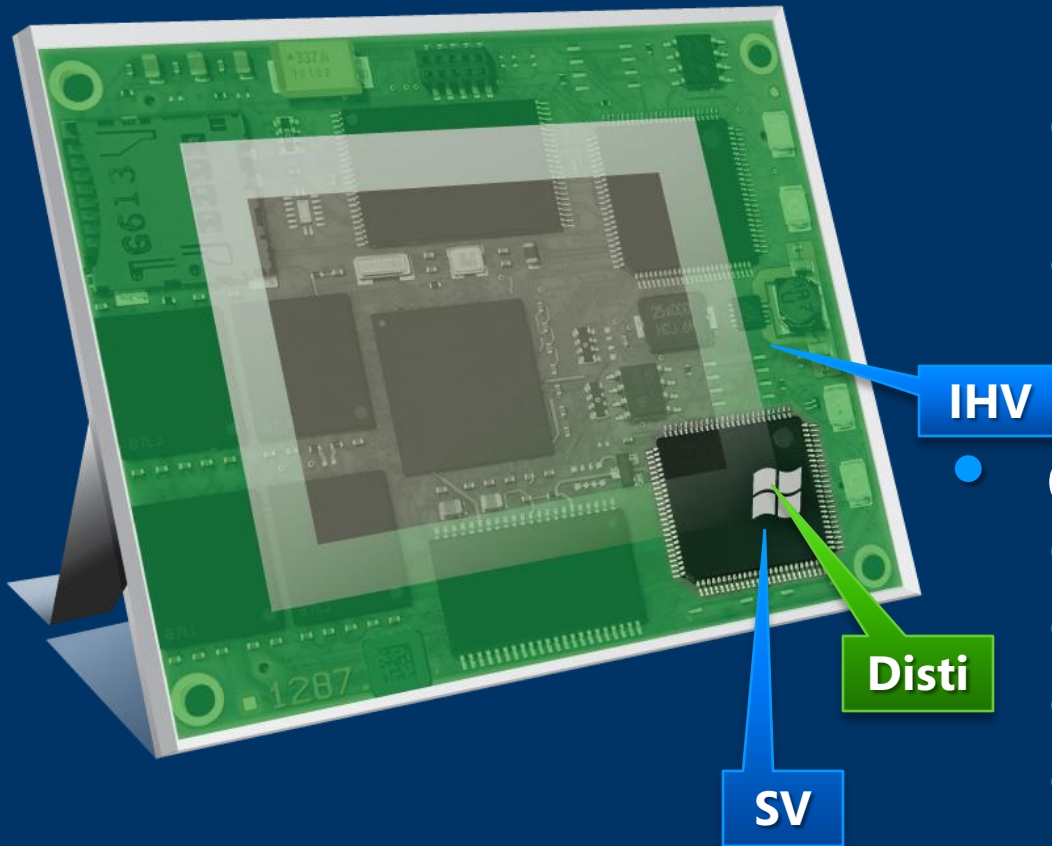
- **SV**
 - A Semiconductor Company
 - Provides embedded device platform and application processors
 - "Silicon Chip"
- Company examples
 - Texas Instruments
 - NXP
 - ST Microelectronics
 - Samsung
 - Freescale

Independent Hardware Vendor



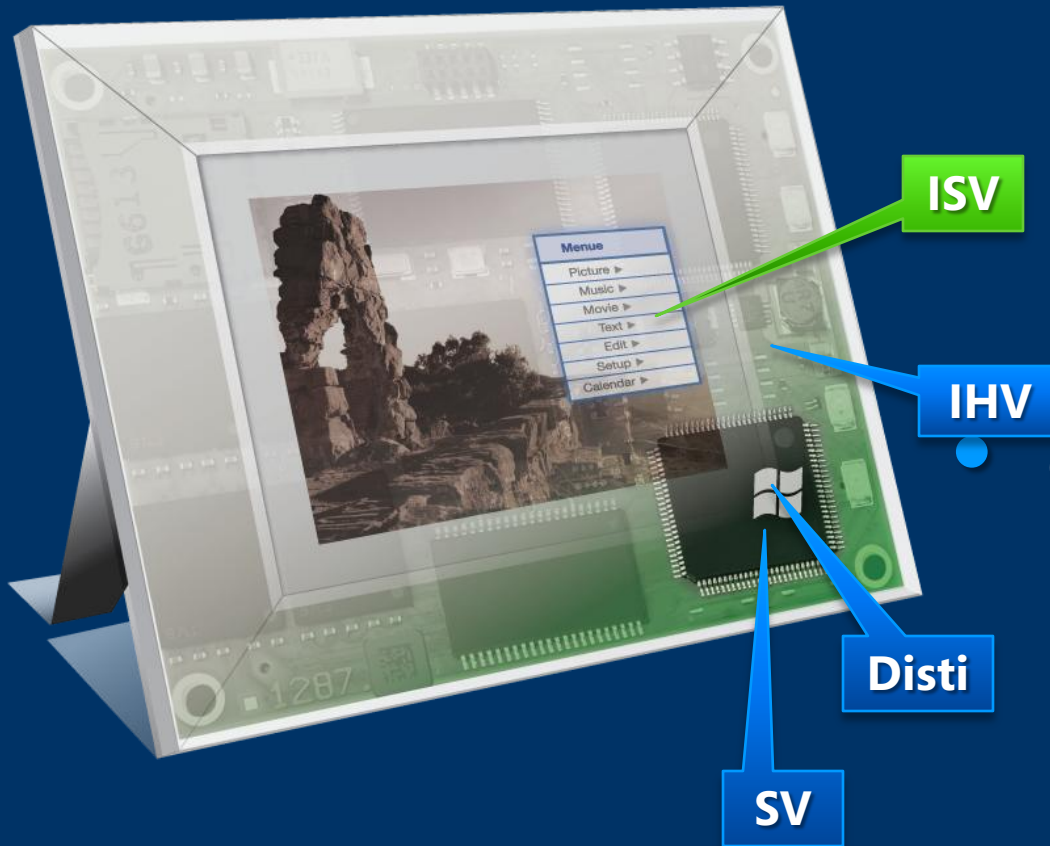
- **IHV**
 - Designs and supplies reference "circuit board"
 - Provides Windows Embedded solutions
 - Hardware
 - Board Support Packages (BSPs)
 - Possibly application integration
- Company examples
 - Phytex
 - Applied Data Systems
 - ICOP
 - Advantech

Distributor



- **Disti**
 - Distributes toolkits, run-time licenses, COAs
 - Customers are OEMs/ODMs
 - Manages relationships and license/contract agreements
- Company examples:
 - Arrow
 - Avnet
 - BSQUARE
 - DST

Independent Software Vendor



- **ISV**
- Provide embedded applications, system level software and tools for developing and debugging
- Company examples:
 - General Software, Adobe, Ardence

System Integrator / Training Partner



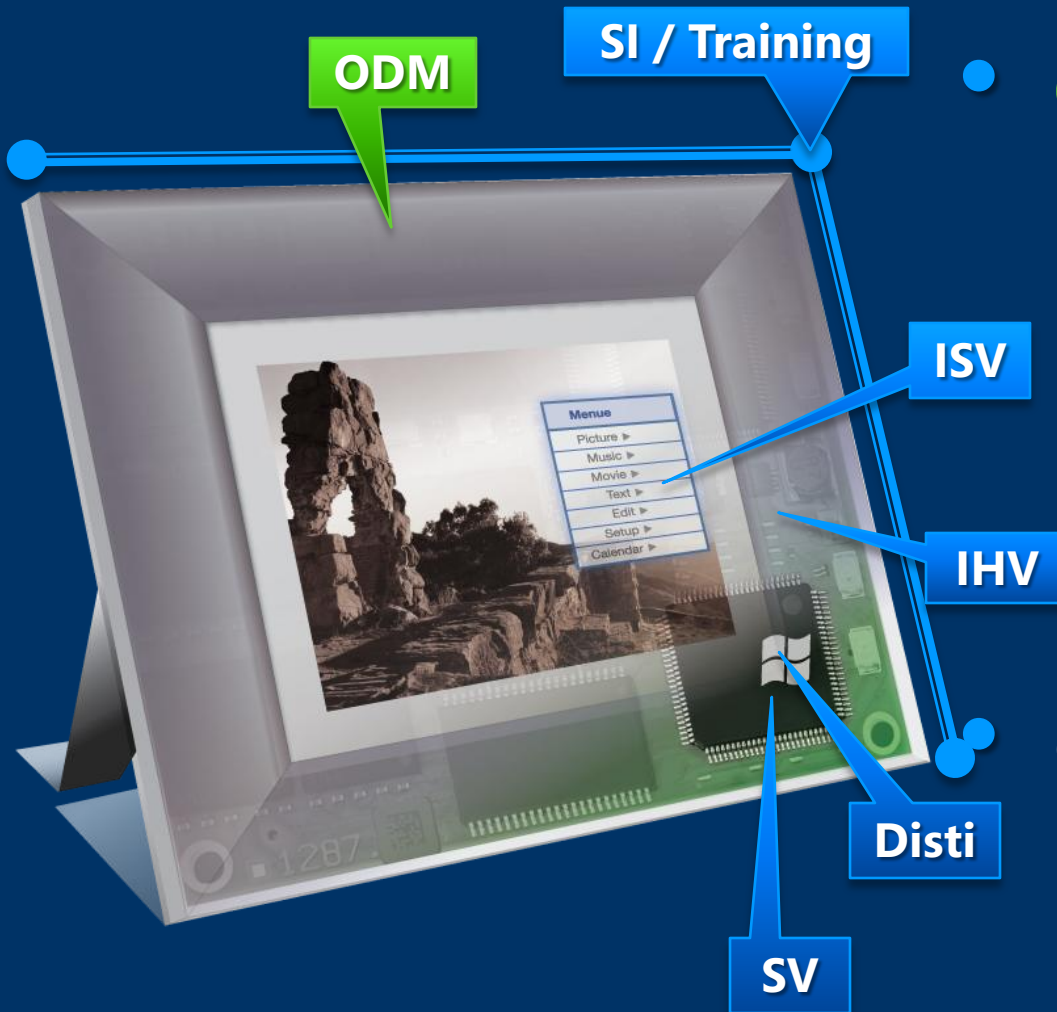
- **SI / Training Partner**

- Either connects at any level in the product design cycle
- SI supplies embedded device software integration
- Trainers train on Windows Embedded OS
- Trainers provide Windows Embedded platform materials and courseware

Company examples:

- Intrinsic
- BSQUARE
- Prevas
- Hilf GmbH

Original Device Manufacturer



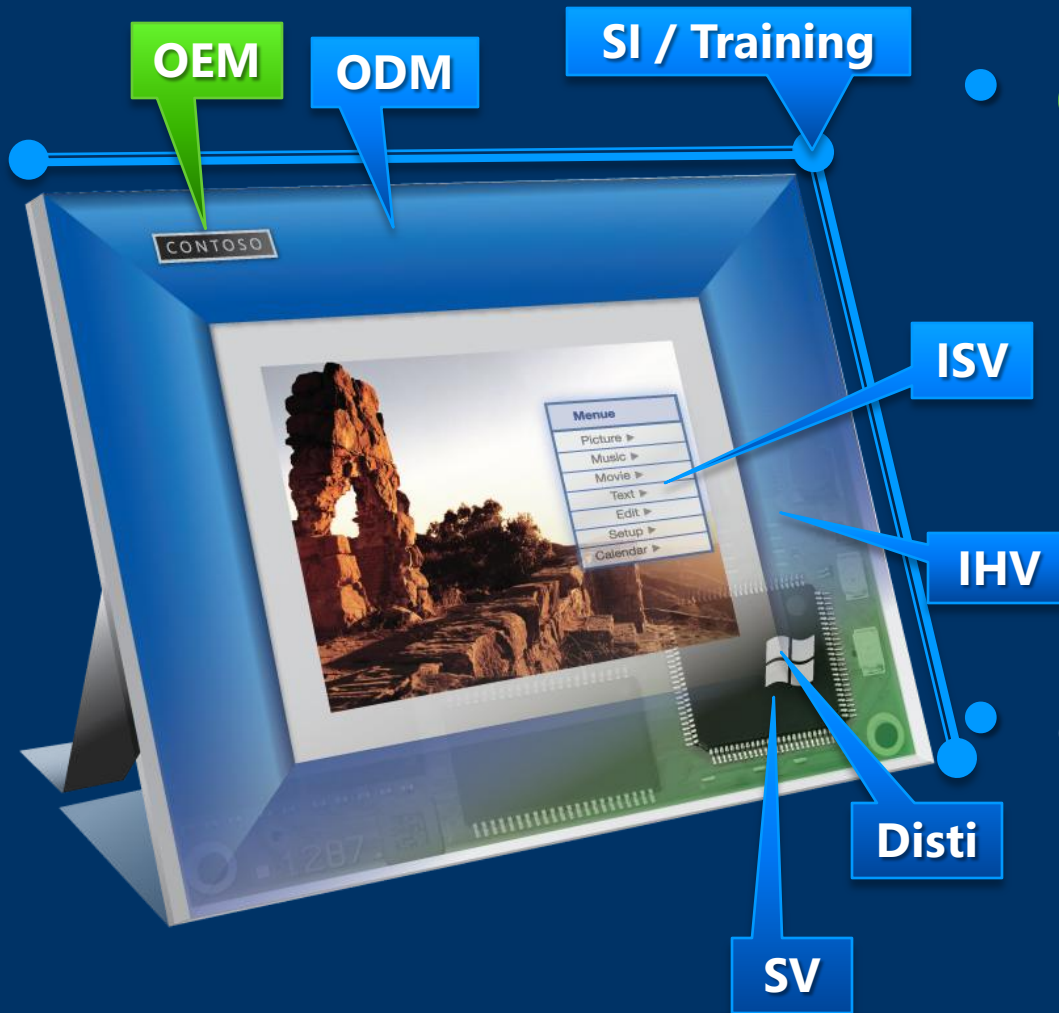
• ODM

- Builds “white label” cases
- Designs and manufactures end products for OEMs
- Diverse product form factors

Company examples:

- Tatung
- Wistron

Original Equipment Manufacturer



• OEM

- Builds and ships to end-user channel
- Produces, markets, and sells products and designs to channel

• Company examples:

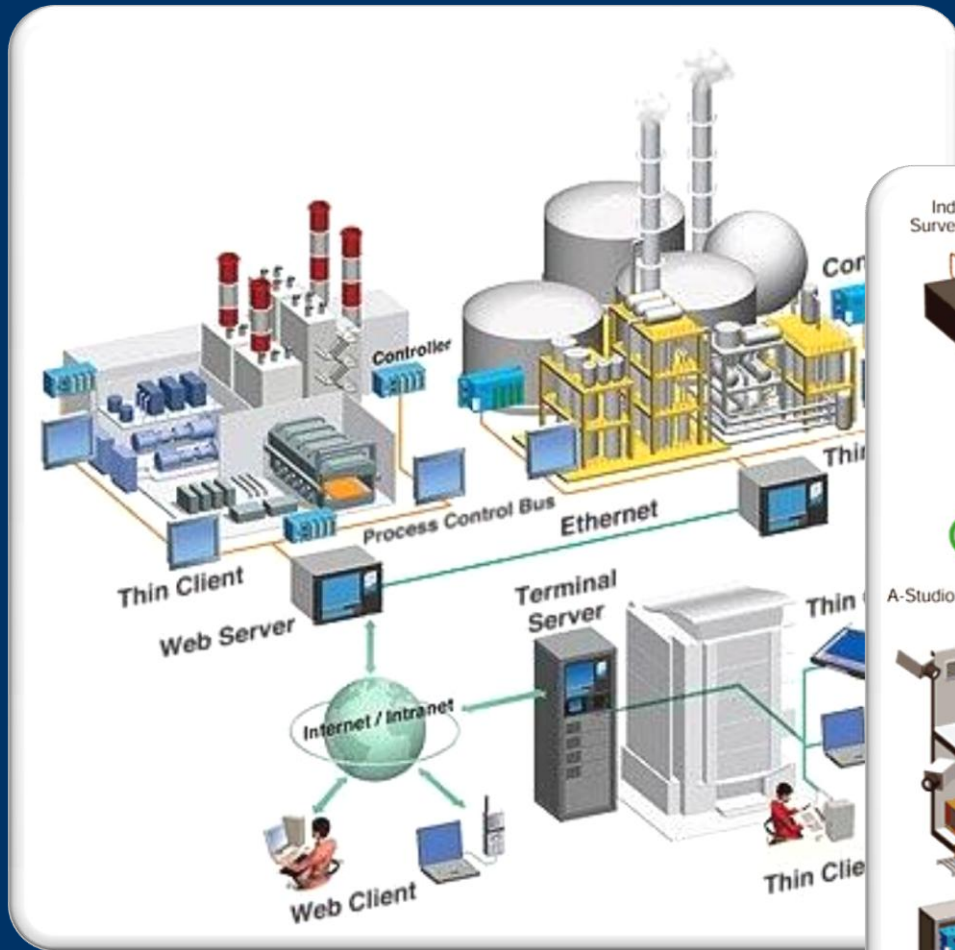
- Toshiba
- Phillips

Who's Who in the Embedded Systems Ecosystem?



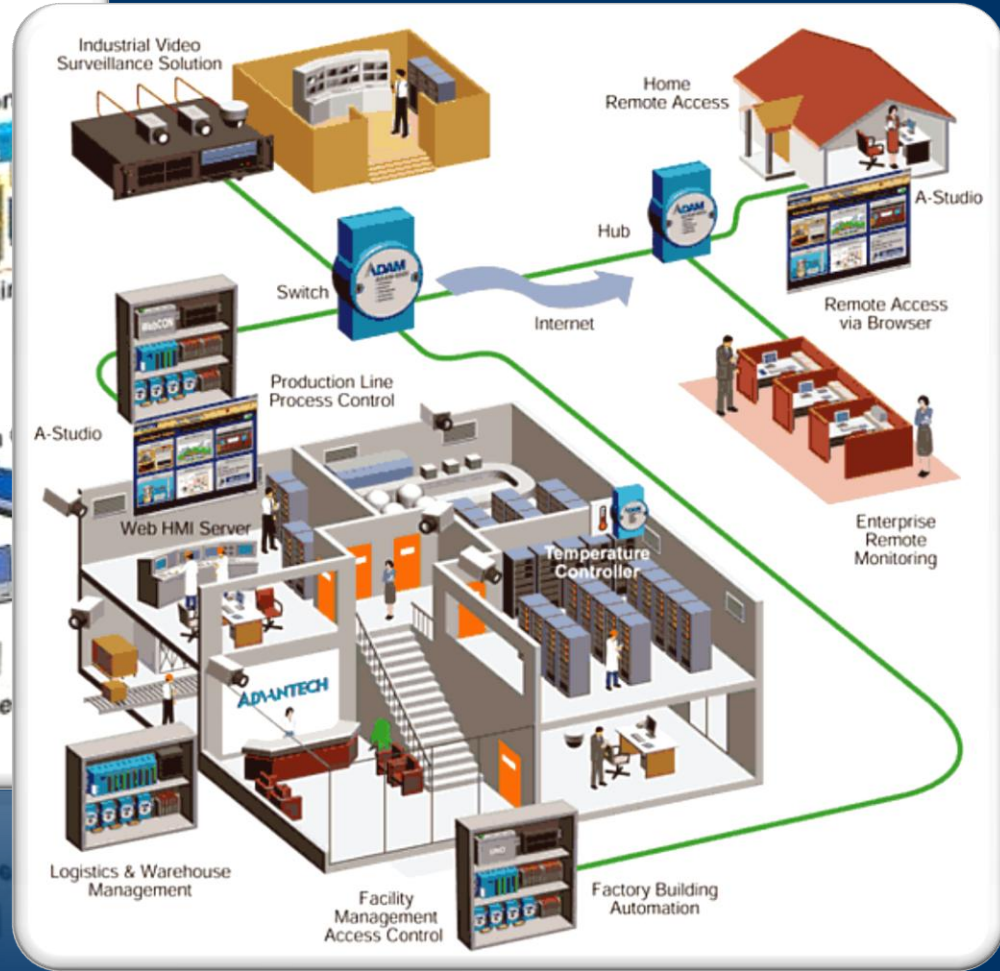
- **Silicon Vendors**
 - Design and build platform processors
- **Independent Hardware Vendors**
 - Develop hardware solutions, BSPs, & AI
- **Distributors**
 - Sell and license Windows Embedded platforms
- **Independent Software Vendors**
 - Develop applications and system level software
- **System Integrators**
 - Embedded device software integration
- **Training Partners**
 - Deliver training, materials & courseware
- **Original Device Manufacturer**
 - Design and manufacture products for OEMs
- **Original Equipment Manufacturer**
 - Produce, market, and sell to end customers

Business Scenarios



Industrial Process Control

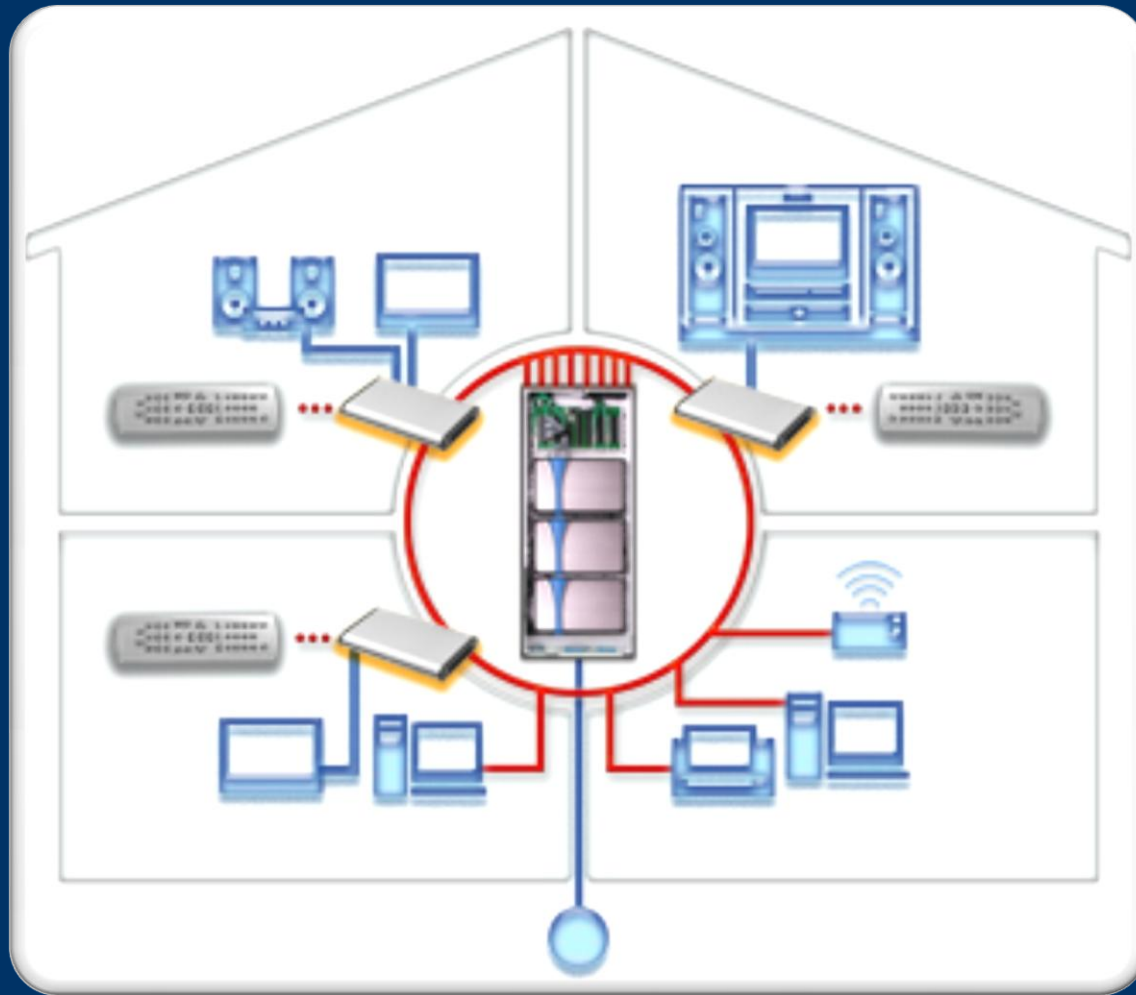
Factory Automation



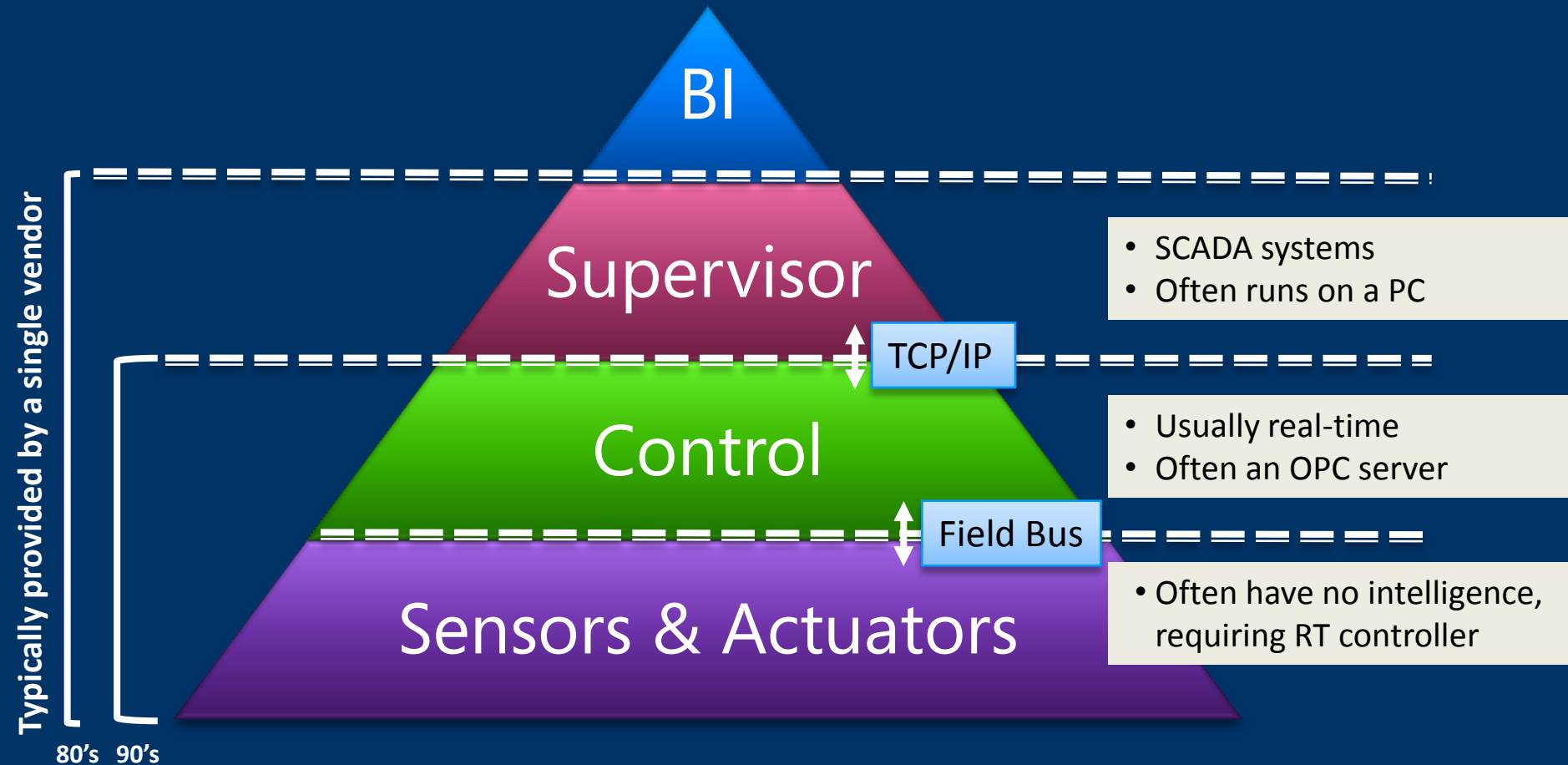
Access Control
Management
Factory
Building
Automation

Consumer Scenario

Home Automation

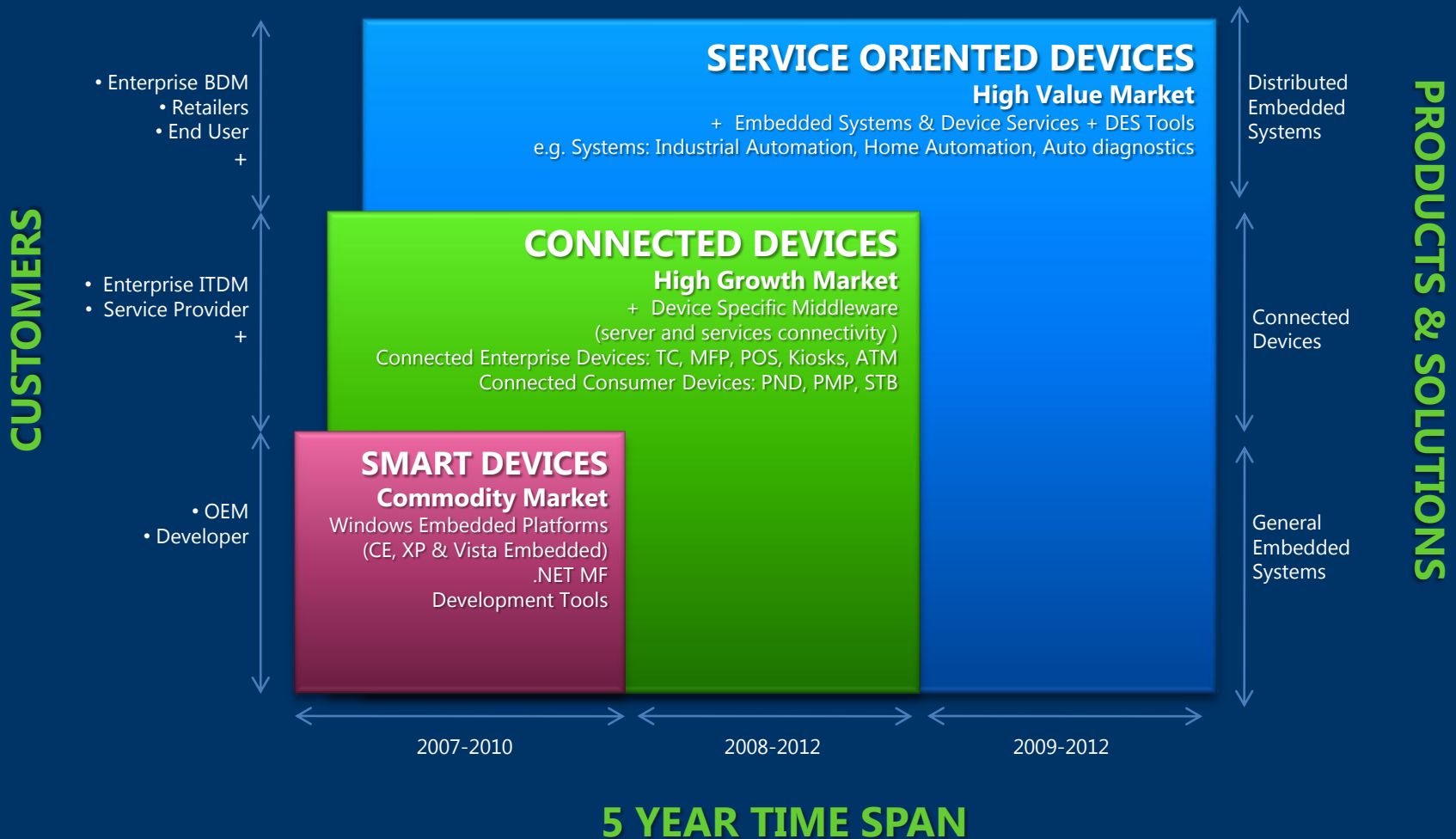


Generalized DES Model

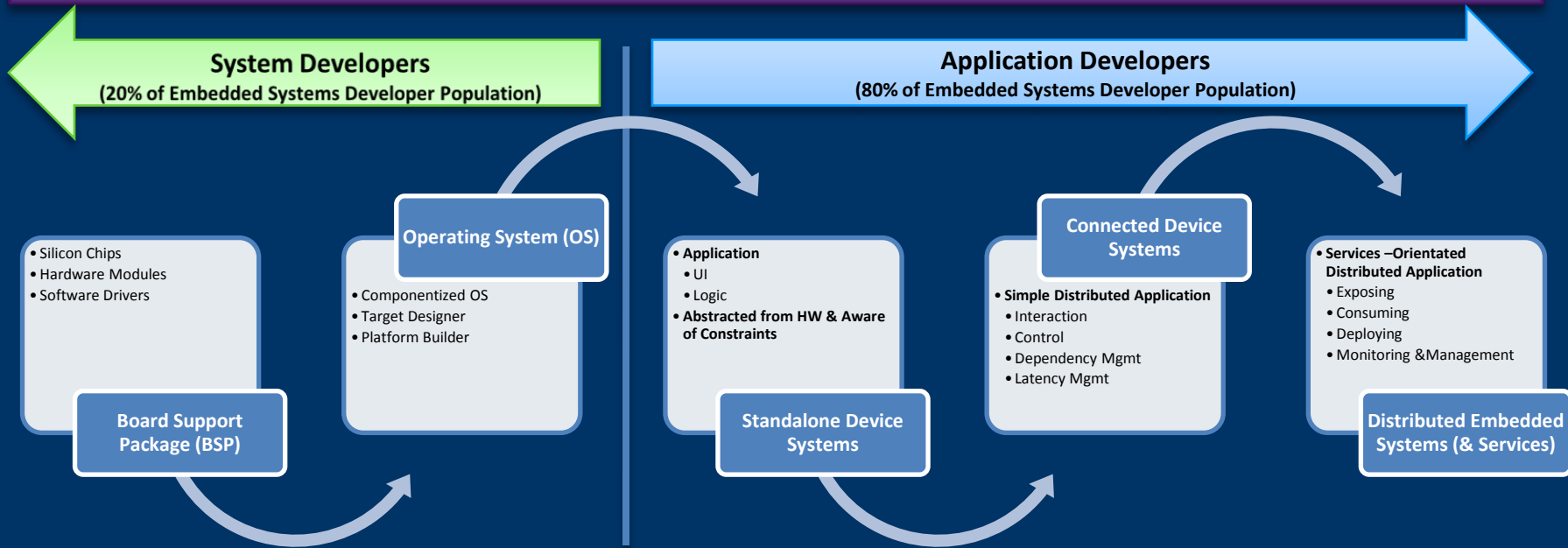


SCADA: Supervisor Control and Data Acquisition

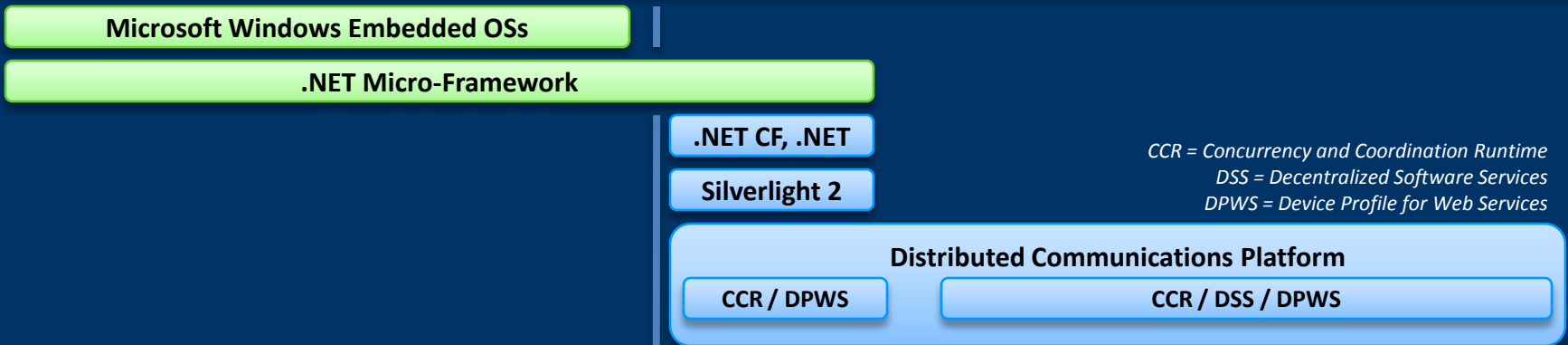
Market Opportunity Service Oriented Devices



Embedded Systems Development Lifecycle



Applicable Technologies on the Microsoft Platform



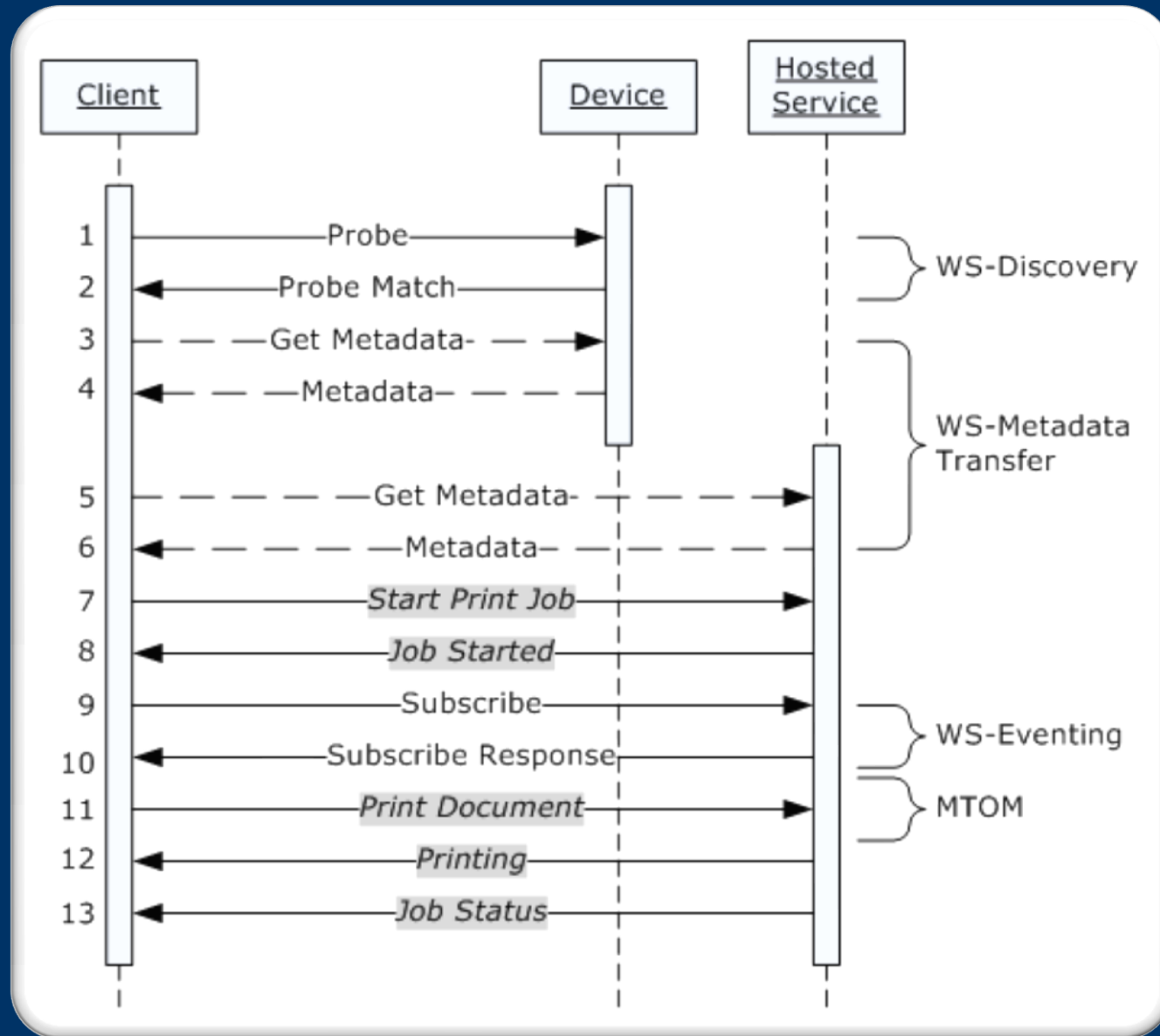
Tools



Devices Profile for Web Services (DPWS)

- Public Specification
 - Contributed to by Intel, Microsoft, Lexmark, Ricoh
 - Included under the Microsoft Open Specifications Promise
 - Web services-based plug & play functionality for resource-constrained devices
 - Extensible, but defines a set of implementation constraints
- Platform Independent
 - Messaging (SOAP 1.2, WS-Addressing, MTOM, XML, XSD)
 - Transport (HTTP/1.1 and UDP bindings)
 - Service discovery (WS-Discovery), Service description (WSDL, WS-Policy, WS-Transfer), Service Event Subscriptions (WS-Eventing)
 - Common vocabulary
- OASIS Web Services Discover and Devices Profile (WS-DD) TC recently formed to define a standard based on WS-Discovery, SOAP-over-UDP, and DPWS
 -
- Microsoft implementations - Web Services on Devices API (WSD-API)
 - Windows Vista, Windows Server 2008, Windows CE 6.0 R2 (native C++/COM)
 - .NET Micro Framework DPWS (managed library)

Example DPWS Interaction - Printer



Communications for Distributed Embedded Systems

Microsoft Robotics Developer Studio

Not just for robotics applications!

Runtime

- Coordination and Concurrency library (CCR)
- Distributed Services Framework (DSF)

Authoring Tools

- Visual Simulation Runtime and Editor
- Visual Programming Language (VPL)
- Visual Studio templates

Services

- Samples and tutorials
- ~~• Robot services~~
- ~~• Robot models~~
- Infrastructure services

MRDS Runtime

**Coordination and
Concurrency library
(CCR)**

- **Multi-Core / Many-Core**
- Asynchrony
- Concurrency
- Resiliency

**Distributed Services
Framework
(DSS)**

- **Distributed Computing**
- Composability
- Loose coupling
- Decentralization

CCR

CCR

Concurrency and Coordination Runtime

Message-oriented programming model,
optimized for:

- Asynchrony (*queuing*)
- Concurrency (*scheduling*)
- Coordination (*arbitration*)
- Resiliency (*failure handling*)

CCR Key Concepts

Message



data orientation

Port



Receiver

```
delegate (int n)
{ /*handler*/ }
```

queuing

Arbiter



arbitration

Dispatcher Queue

Task



```
{ /*handler* }
```

scheduling

Dispatcher



CCR Key Concepts - Setup

Port



```
Port<int> p = new Port<int>();
```

3

A Receiver

```
delegate(int n)  
{ /*handler*/ }
```

```
Arbiter.Activate(q,  
    Arbiter.Receive(true, p,  
        delegate(int n)  
        { /*handler code...*/ }));
```

4

Dispatcher Queue



```
DispatcherQueue q =  
    new DispatcherQueue("", d);
```

2

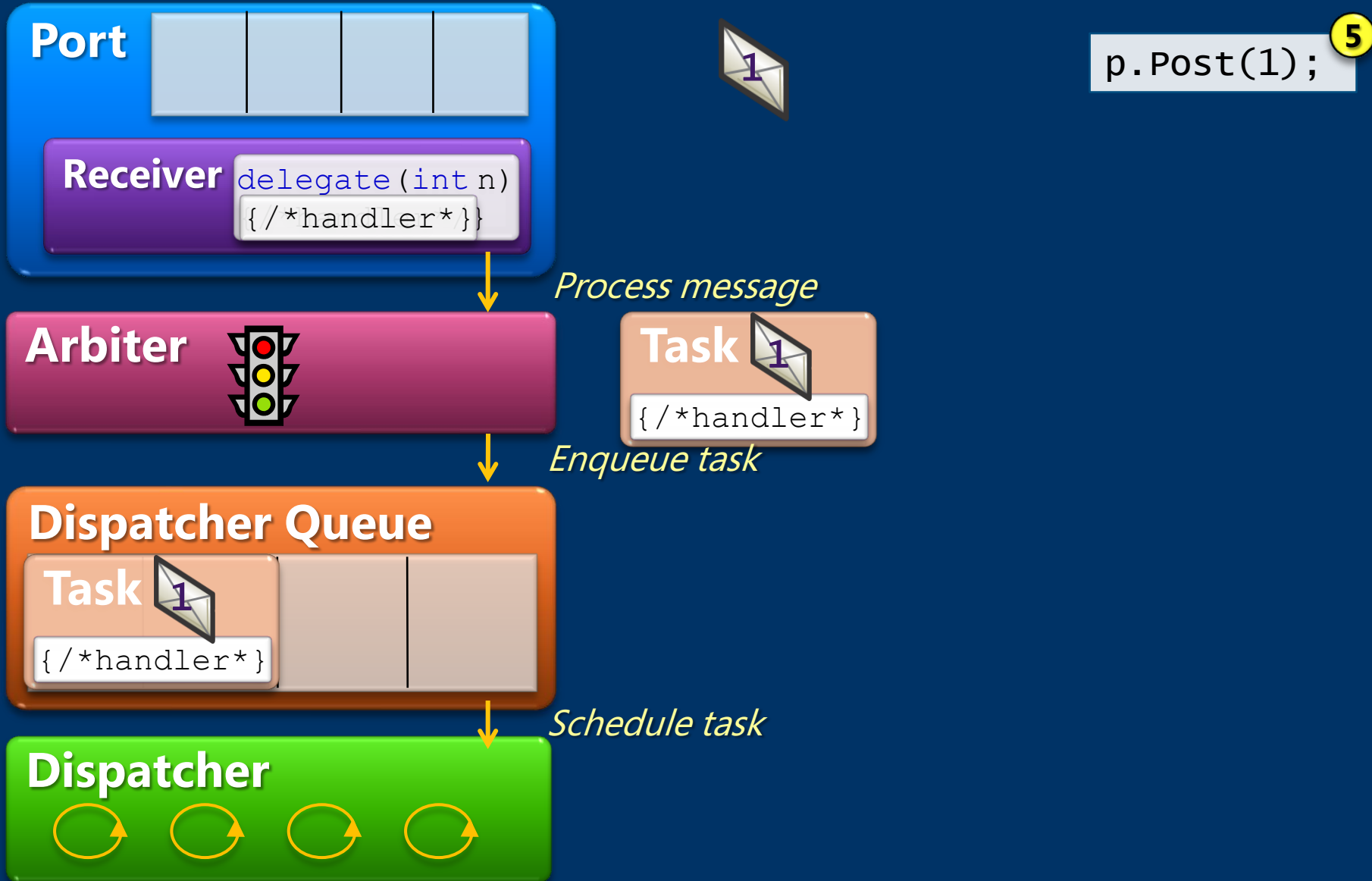
Dispatcher



```
Dispatcher d =  
    new Dispatcher(0, "");
```

1

CCR Key Concepts - Execution



CCR Coordination Primitives

Exposed via *Arbiter* methods

Code-Scheduling
(non port-specific)

FromHandler

FromIterator
Handler

Single-Port
Primitives

Single Item
Receiver

Multi-Item
Receiver

Interleave
(Reader/Writer)

Multi-Port
Primitives

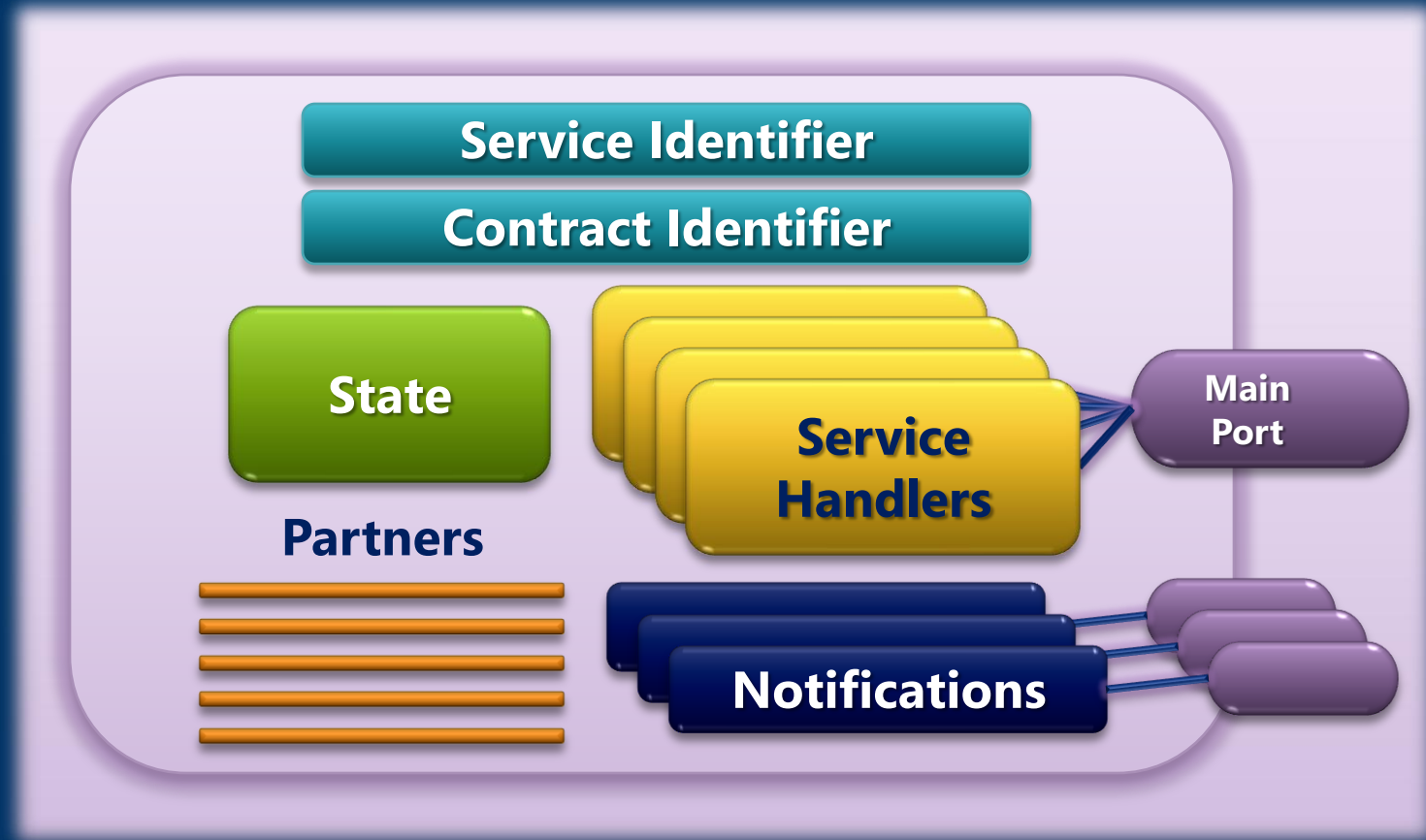
Join
(Logical AND)

Choice
(Logical OR)

Multi-Port
Receiver

DSS

DSS Architecture



Services as Units of Orchestration

- Service Properties
 - Identity (URI)
 - Structured State
 - Composition through partnering
- Uniform Behavior
 - State retrieval and manipulation
 - Service creation & Termination
 - Notifications are coupled to state changes



Builds on REST and HTTP

DSSP

Structured Data Manipulation - INSERT, UPDATE, ...

Events represent state changes

Application graph observable

HTTP

Simple, uniform APIs - GET, POST, ...

UI Separated from Implementation

Session-less

Microsoft Open Specification Promise

Anatomy of a DSS Service

<http://host:port/servicepath>

Service Properties

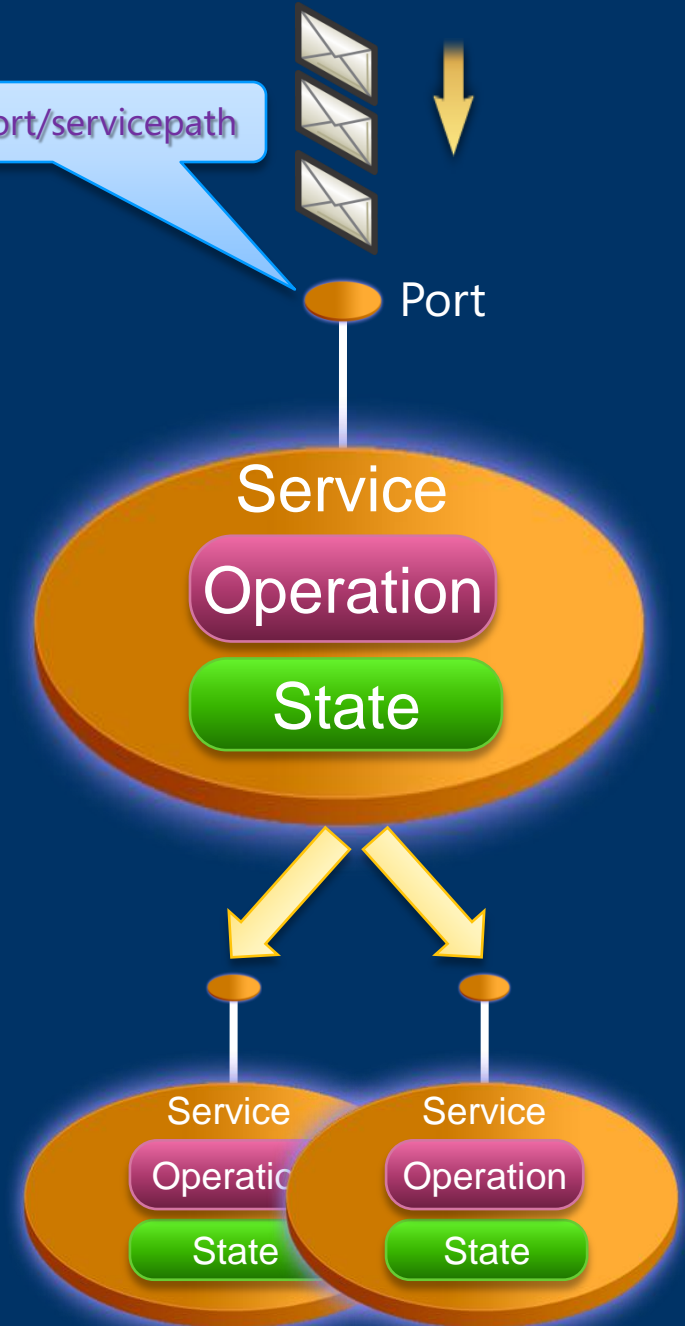
- Service URI
- Contract URI
- Structured state
- Partner services

Service Contract

- Observable state
- Supported DSSP operations

Services can..

- Implement a data contract
- Extend a data contract
- Implement multiple data contracts

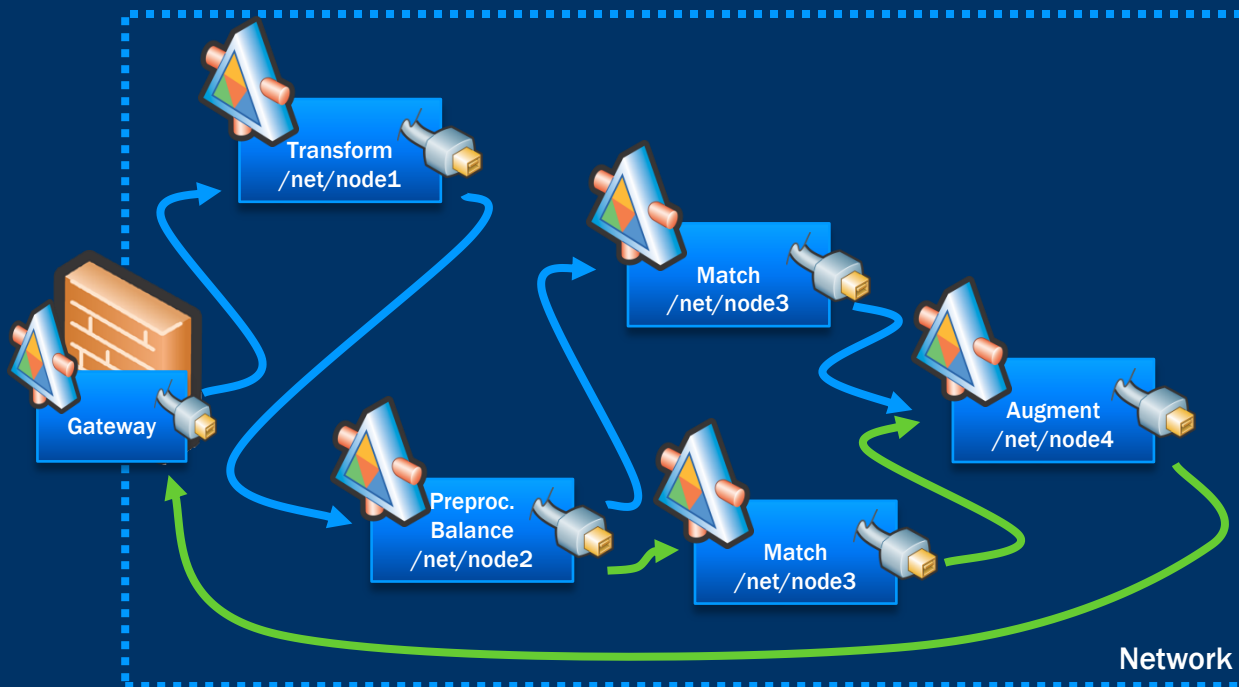


Fabrig4Dss

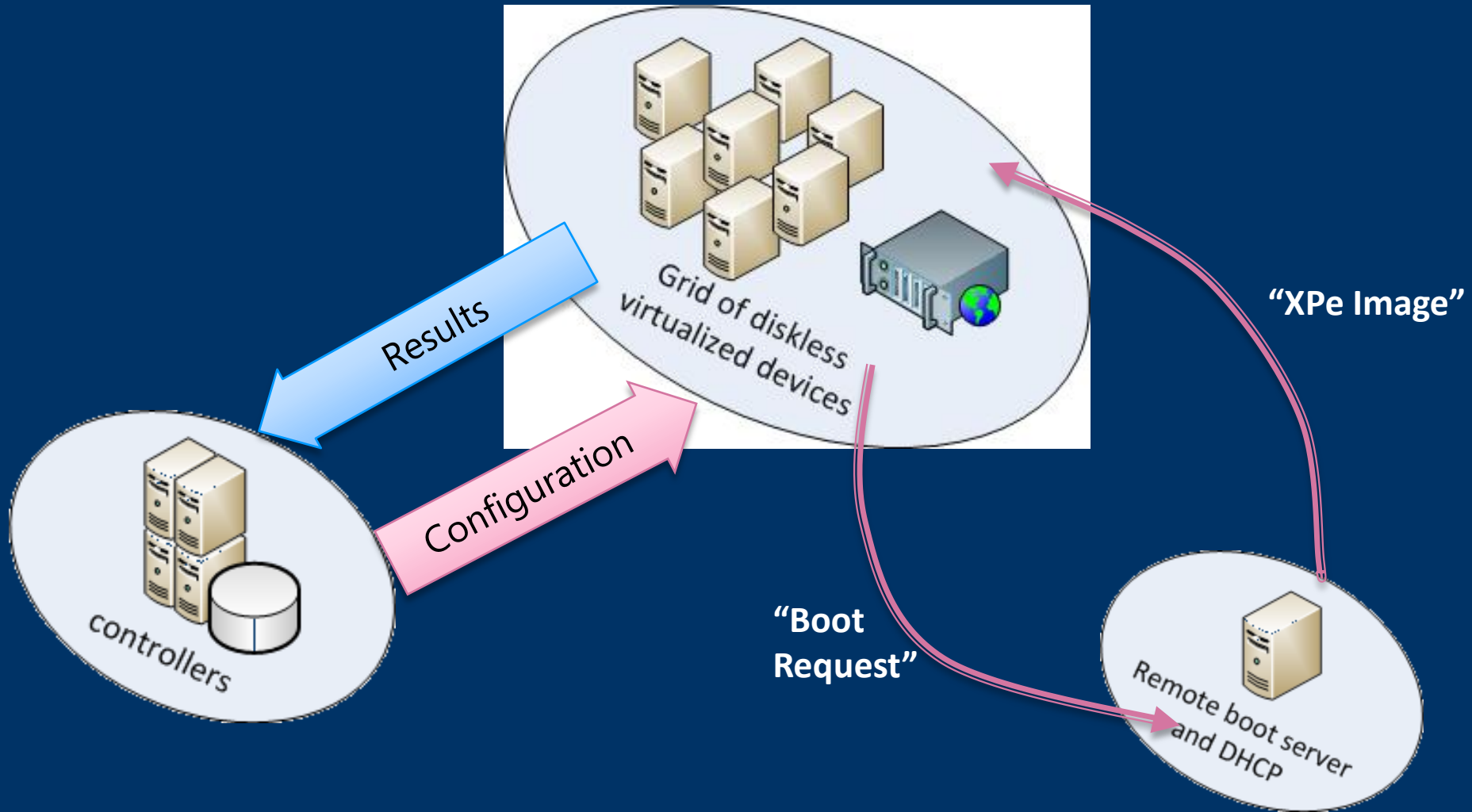
CCR/DSS Application Example

Fabrig4Dss - Queuing Networks

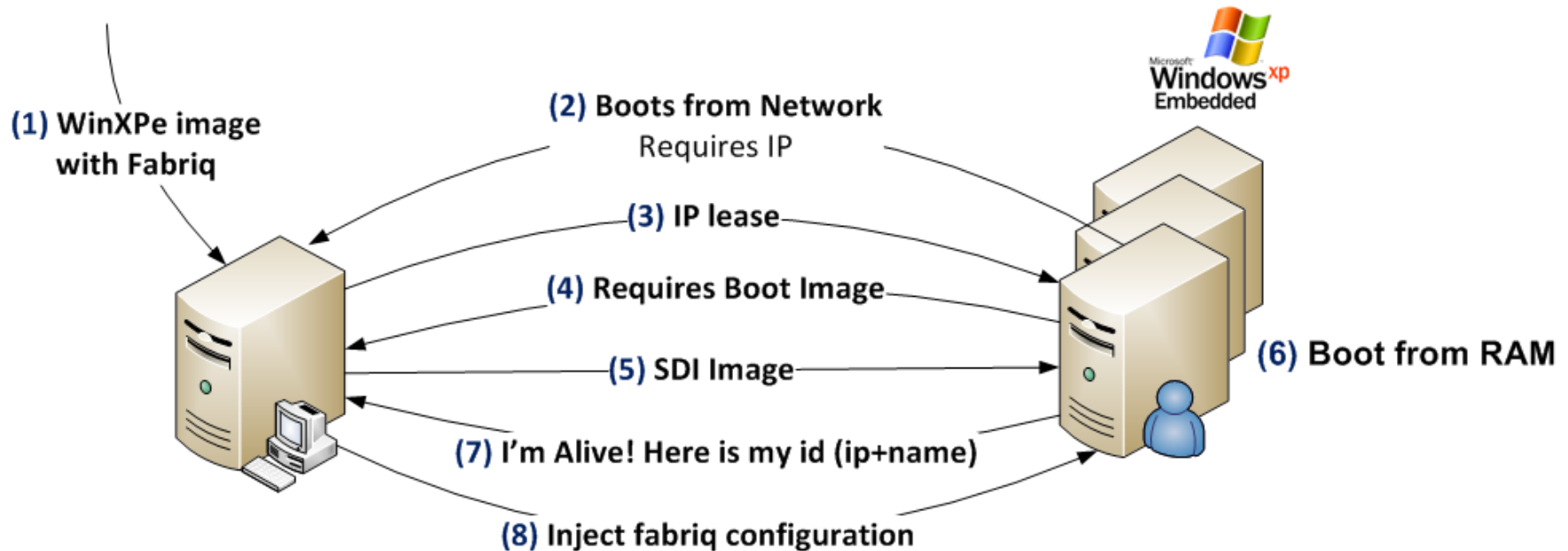
- A "network" is a set of connected pipelines
 - Each message travels along a path in the network
 - Path is determined dynamically by routing rules
 - A path is a pipeline made from processing units
- Processing units in a network are called "nodes"



A Virtualized Diskless Grid



Remote Boot Process



Name: fabriqlab-server01
Role: Controller
Remote Boot Server
HW: VPC (256mb)
Soft:

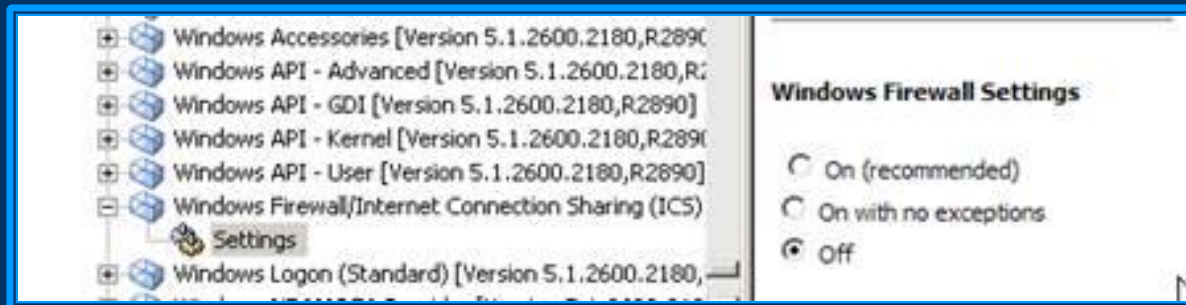
- Win2k3 EE
- DHCP Server
- Remote Boot Manager
- SDI image uploaded

Name: fabriqlab-client01
Role: Grid Node
HW: VPC (1024mb - Diskless)
Soft:

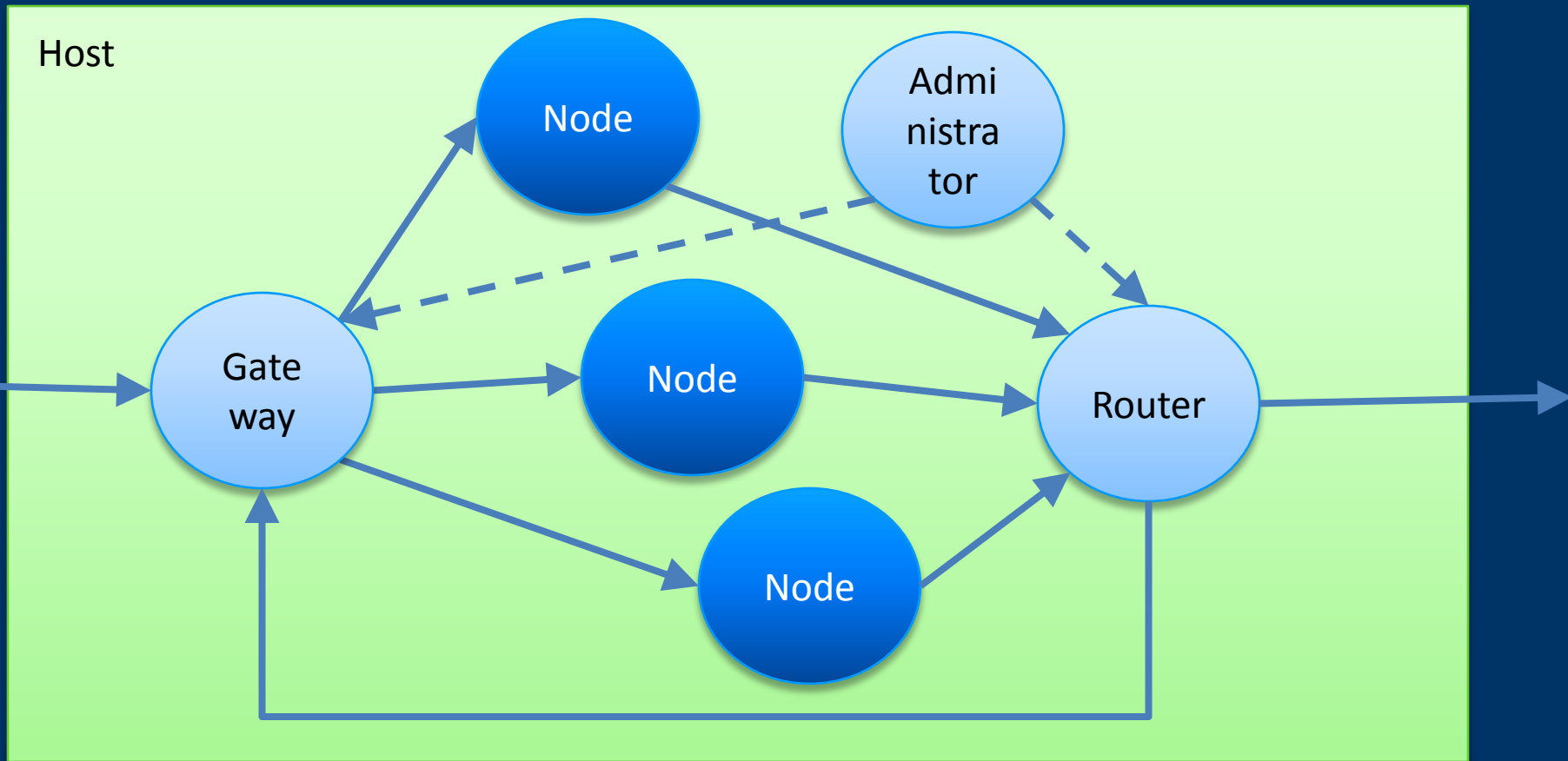
- NET 2.0
- Robotics Studio 1.5
- Fabriq

Windows XP Embedded

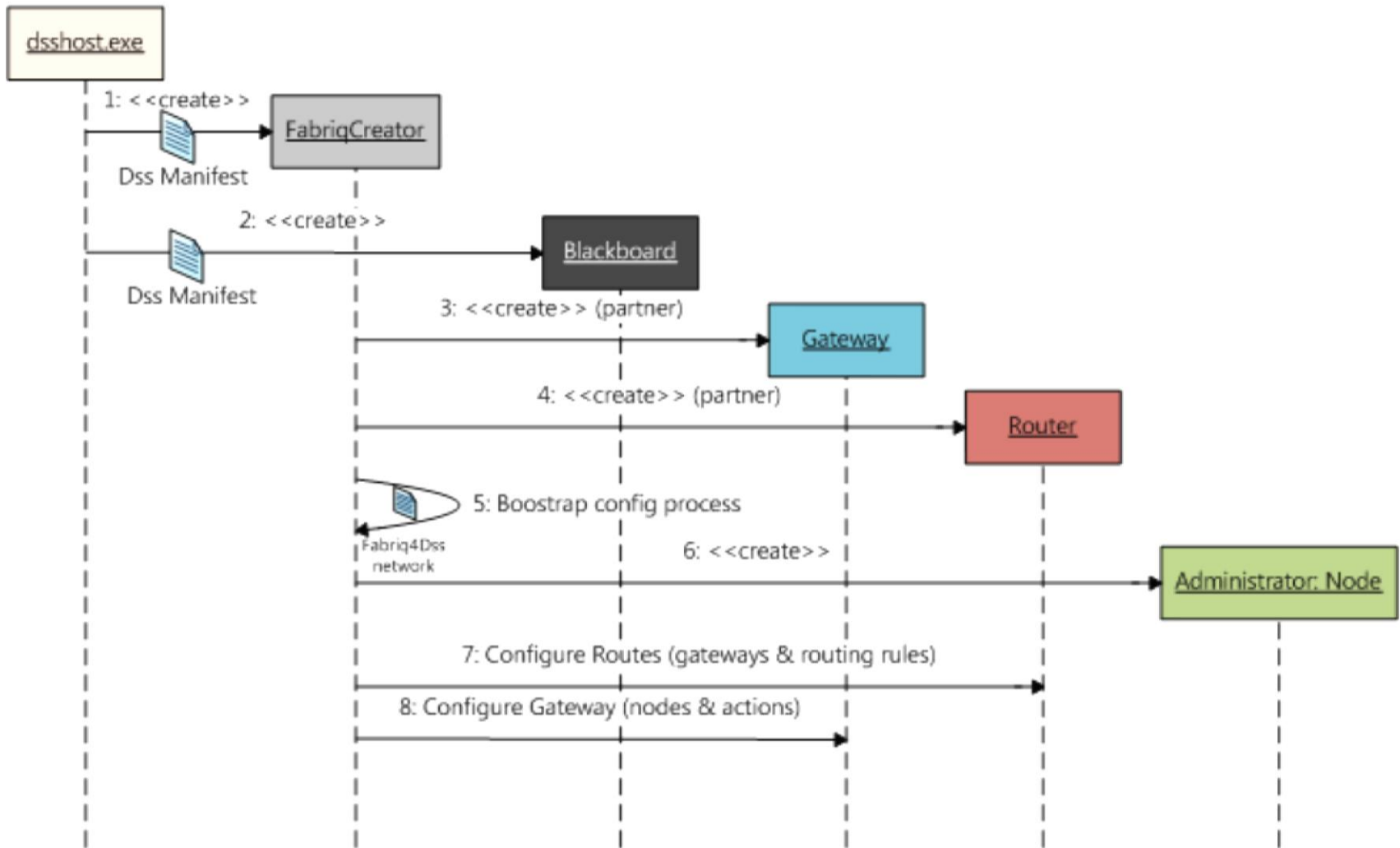
- Current XPe image size is ~450 Mb
 - Takes 6 minutes to boot into a Virtual PC host running on a Windows Server 2008 machine
 - Unicast streaming, but could use multicast in large-scale deployments
 - Generates unique SIDs and machine names on each boot
 - Image customized using Target Designer to minimize size, and incorporate user components



Building a Grid Processing Node



Bootstrapping the Fabriq Grid



Anatomy of Grid Nodes

Suscriptions (A)

- New Configuration (from Configurator)

Events (A)

- Via Application Configuration

Operations (A)

- Get Host State (aggregate of all node states)
- Via Application Configuration

Operations (C)

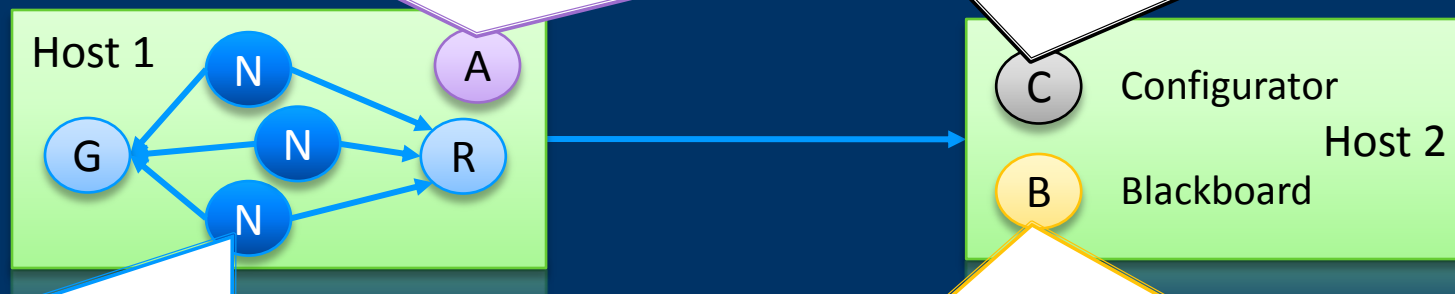
- New Configuration

Events (C)

- New Configuration

Suscriptions (C)

- New Node
- Dropped Node
- Configuration Deployed



Operations (N)

- Process Message

Events (N)

- Instrumentation
- Data Node Changed

Suscriptions(N)

- None

Operations (B)

- Get Key/Value
- Set Key/Value
- Add Value to Key
- Get Values of Key

Events (B)

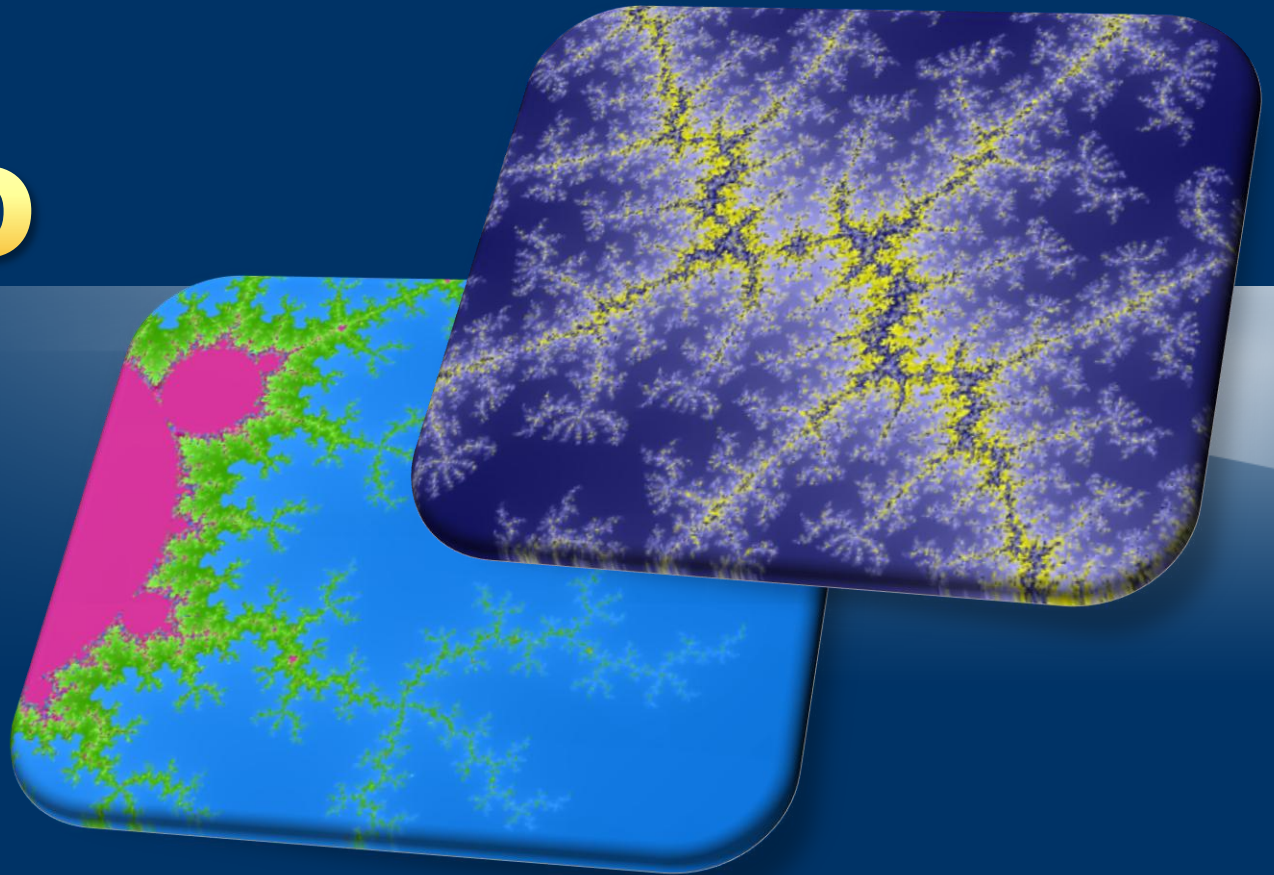
- Key/Value Changed

Suscriptions (B)

- New Node
- Dropped Node
- Configuration Deployed

Map-Reduce Fractals

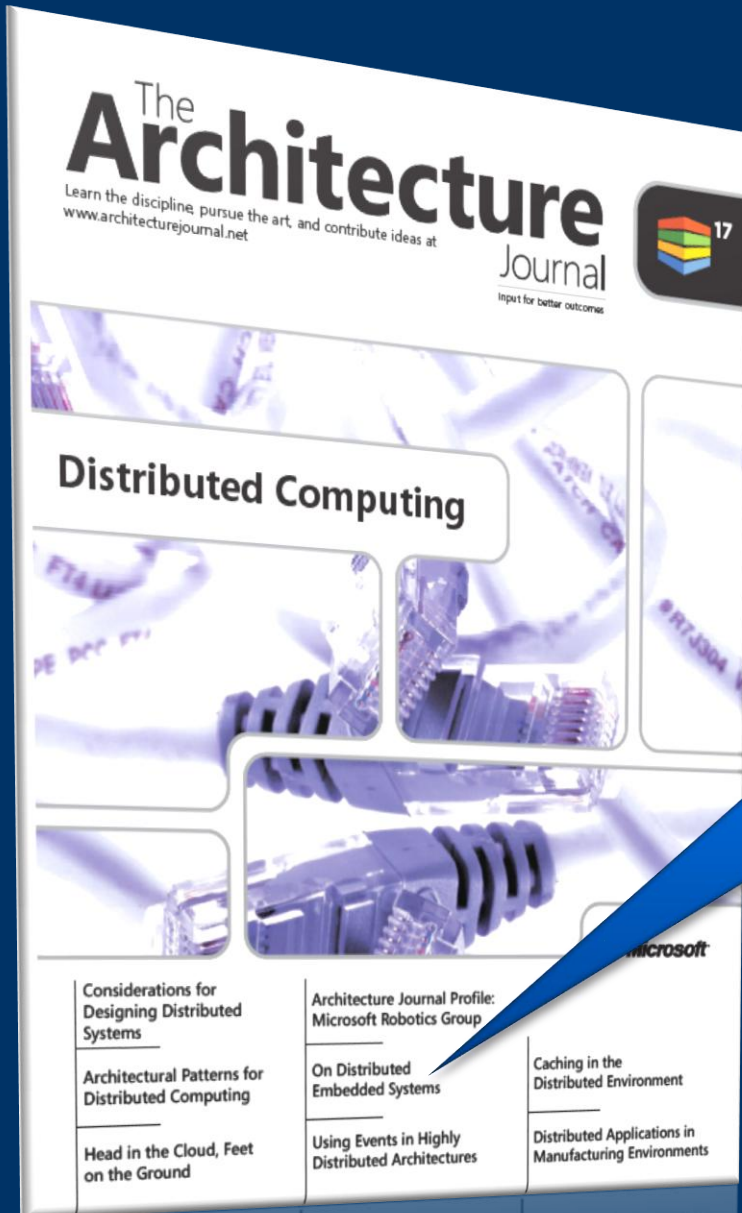
demo



```

<?xml version="1.0" encoding="utf-8" ?>
- <Fabriq4Dss configuration="FabriqFractals" version="1.0">
  <handlerType name="calculate" class="FabriqFractals.Handlers.CalculateHandler, FabriqFractals" />
  <handlerType name="getvalue" class="Robotics.Fabriq4DssTools.MessageHandlers.GetValueHandler, Fabriq4DssTools.Y2008.M05" />
  <handlerType name="logkeyvalue" class="Robotics.Fabriq4DssTools.MessageHandlers.LogKeyValueHandler, Fabriq4DssTools.Y2008.M05" />
  <handlerType name="setkeyvalue" class="Robotics.Fabriq4Dss.Runtime.Blackboard.SetKeyValueHandler, Fabriq4Dss.Y2008.M04" />
  <handlerType name="setconfiguration" class="Robotics.Fabriq4Dss.Runtime.Blackboard.ConfigurationHandler, Fabriq4Dss.Y2008.M04" />
- <nodeType name="calculator">
  - <actions>
    - <action name="calculate" match="urn:calculate">
      - <pipeline>
        <handler type="calculate" />
      </pipeline>
    </action>
  </actions>
</nodeType>
- <nodeType name="blackboard">
  - <actions>
    - <action name="setkeyvalue" match="urn:SetKeyValue">
      - <pipeline>
        <handler type="setkeyvalue" />
      </pipeline>
    </action>
    - <action name="setconfiguration" match="urn:SetConfiguration">
      - <pipeline>
        <handler type="setconfiguration" />
      </pipeline>
    </action>
  </actions>
</nodeType>
- <network name="FabriqFractalsNetwork">
  <node name="blackboard" type="blackboard" host="localhost:50001" />
  <node name="calculator" type="calculator" host="localhost:50001;localhost:50003" />
</network>
</Fabriq4Dss>

```



On Distributed Embedded Systems
by Arvindra Sehmi

Summary

- Great opportunities in DES for “traditional” systems and application developers
- CCR/DSS – not just for robotics applications!
- Addresses common challenges
 - Concurrency and asynchrony addressed via coordination
 - Composability (Partnership) is a first-class citizen
- Web/document-centric model
 - RESTful treatment of service interactions
 - State observability of running services
 - Loose coupling

Appendix: Getting Started with CCR/DSS

- Download the product
 - Download Visual Studio if needed
- Watch video tutorials
- Read through CCR documentation
 - Try out concepts in code
 - Use Reflector liberally
- Read DSSP specification
- Read through DSS documentation
- Work through Services Tutorials
- Work through Hosting Tutorials
- Watch robotics and CCR videos on Channel9
- Use online forums for questions

Appendix: Resources

<http://microsoft.com/robotics>

- Product information
- Online Documentation
- Downloads
- Discussion Forums
- Community
- Tutorials
- Videos
- Blogs



Microsoft[®]

Your potential. Our passion.[™]

© 2008 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.
MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.