Dataplane Processing Platform

--- a software perspective

Device Software Solutions for Telecom Platforms

张永军　frank.zhang@enea.com
Business Development Manager

Agenda

- Enea 简介
- Dataplane Processing Platform
  - OSEck RTOS for DSP
  - dSPEED – DSP 集群软件管理平台
- 总结

Volumes……

- Enea’s software contained in 50% of all world base stations.
- Enea’s software contained in 400 million new mobile handsets in 2008
- 50 percent of all mobile phone calls are processed through Enea software

Enea 公司简介

网络和终端软件的领导供应商

- Enea 成立于 1968
- 1989 年在瑞典上市
- 750+ 员工

全球性的公司

- 瑞典(R&D)、美国(R&D)、法国(R&D)、罗马尼亚(R&D)
- 英国(R&D)、中国(R&D) 、德国、以色列、日本等

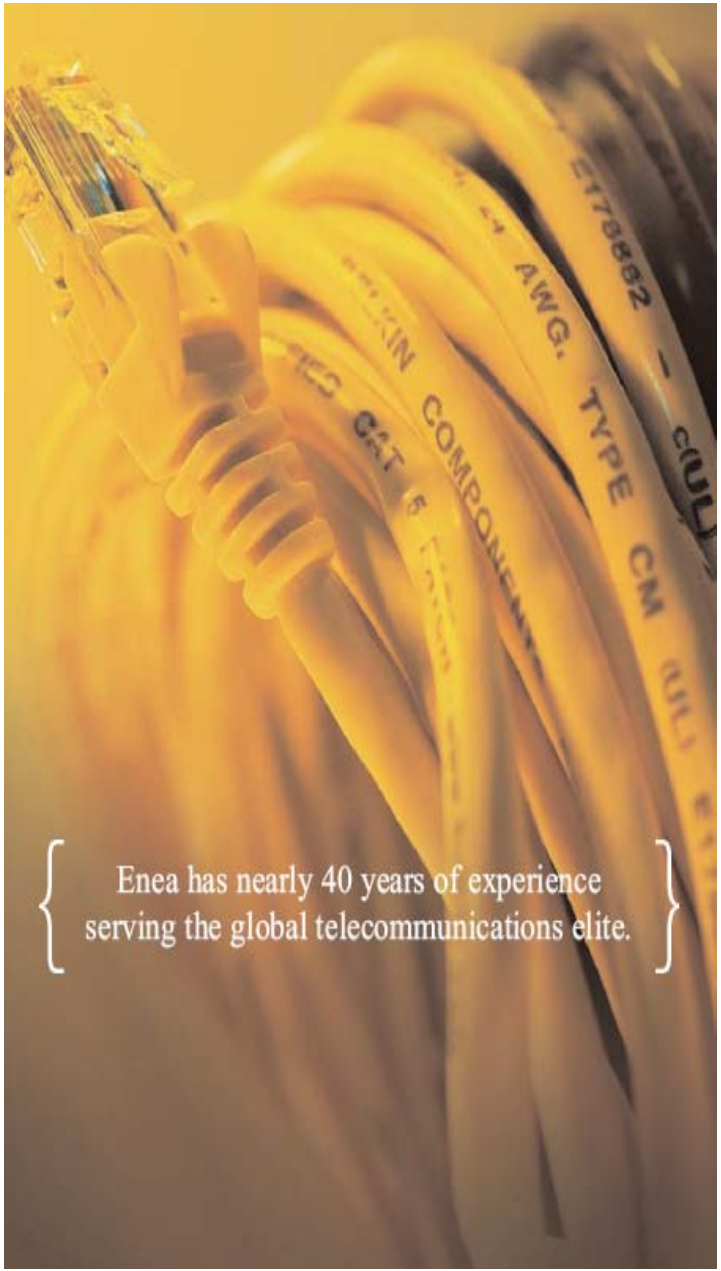
提供独特的软件技术

- 使用在全球上十亿的手机中
- 全球 50% 的基站市场份额
- 2008年销售的手机里 4 亿个带有Enea的软件
- 网络设备、手机、媒体网关、软交换等.....
- 专业咨询服务

重要客户

- Ericsson, Nokia, Alcatel-Lucent, Motorola, Fujitsu, Infineon, Agere, Boeing, Samsung, Sony-Ericsson, Yamaha, Huawei, ZTE, Datang, Potevio, 32所, 615所等…

www.enea.com



Enea 软件产品线

OSE™ (RTOS)

- OSE Delta for RISC, 如 PPC, ARM, XSacle, MIPS
- **OSEck, "compact kernel" 专门为DSP优化设计的内核**
- OSE Epsilon, 内核由100% 汇编编写， 支持MCU 和 SoC
- **Optima: 基于Eclipse的开发工具;**
- **LINX:分布式IPC for OSE, OSEck and Linux**

Polyhedra™

- 性能高、代码小、基于内存运行的实时数据库
- 符合业界标准， 包括 SQL, ODBC, JDBC
- 平台支持 Unix, Linux, Windows 和其它大部分实时操作系统RTOS
- Polyhedra FlashLite：面向于手机市场

Element™

- 应用软件开发和高可用性(HA)中间件构架
- 机箱管理, 故障监控、管理，软件热升级等
- 支持 Linux 和 OSE, possible others

dSPEED ™

- **管理高性能 DSP 群集的商用软件平台**
- **启动和配置，错误处理、监控和监管DSP 群集**
- **事件通知、记录和跟踪，诊断和统计，以及远程调试。**

Netbricks通信协议栈- 基于IP的固网和移动网络Signaling协议: Sigtran, SIP, MGCP, Megaco, H.324m

- 针对数字电路核心网和接入网的Signaling协议: SS7 , ISDN, BLES, RTP, X.25, ..
- 模拟数据传输和接入网: Soft modem, Group 3 fax, POTS, DTMF, Diameter

Accelerator

- 针对融合网络设备的电信级软件基础平台
- 适合 IMS/NGN/IPTV/媒体网关/软交换等 平台

关注于电信

Enea has Trusted Relationships with Customers

ERICSSON

Nokia Siemens Networks

NOKIA
CONNECTING PEOPLE

HUAWEI

FUJITSU

TEXAS INSTRUMENTS

Alcatel·Lucent

ZTE中兴

BOMBARDIER

YAMAHA

Potevio
中国普天

VOLVO

infineon
Never stop thinking

Sony Ericsson

BOEING

SAMSUNG

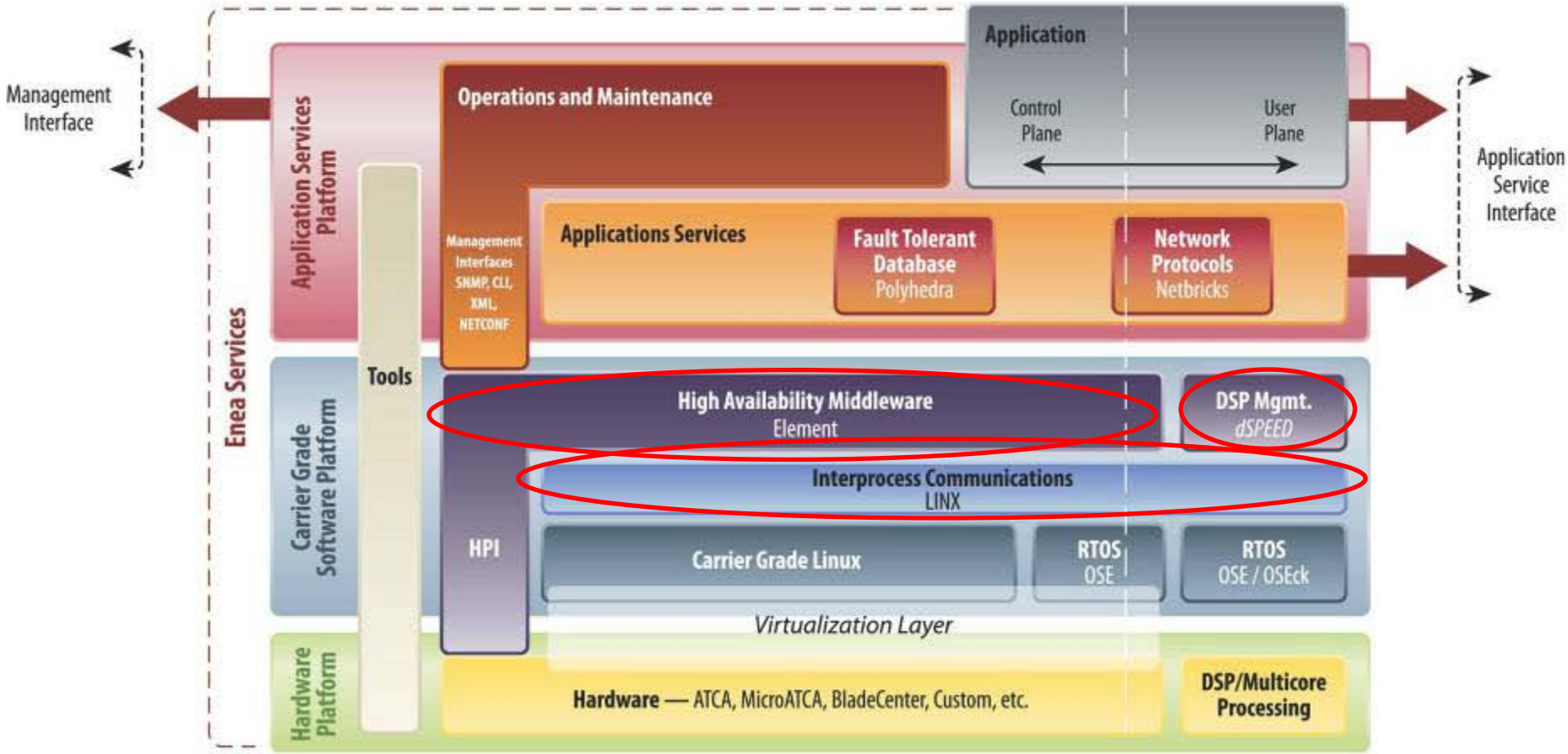
NASA

MOTOROLA
intelligence everywhere

tellabs®

LOCKHEED MARTIN
We never forget who we're working for™

Enea Products Power the CGBP

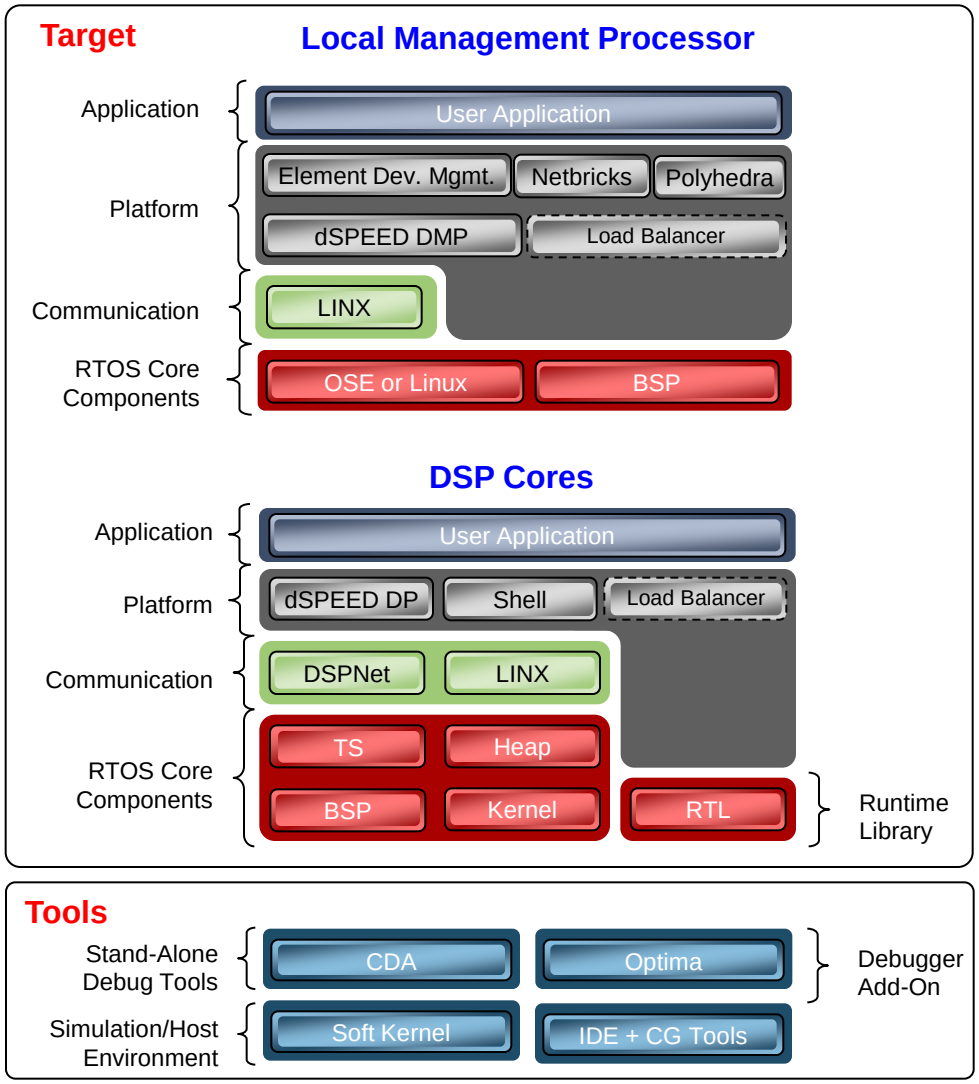


Element, LINX, and dSPEED make up the core of Enea's CGBP middleware offering and work in conjunction with other Enea software products.

Agenda

- Enea 简介
- Dataplane Processing Platform
  - OSEck RTOS for DSP
  - dSPEED – DSP 集群软件管理平台
- 总结

Enea DSP SW Solution



- OSEck - OSE Compact Kernel
  - A DSP-optimized version OSE RTOS
- LINX
  - Transparent, reliable, high-performance interprocess communications services for complex distributed systems
- DSPNet
  - Secure IPv4/IPv6 stack, IPsec and SSH support in a modular design
- Shell
  - Standalone Shell accessible via LINX or DSPNet
- Load Balancer (work in progress)
  - A work scheduler and resource manager for multiprocessing and distributed load balancing
- dSPEED
  - Suite of management, debug and error handling services for developing, deploying, and maintaining DSP “user plane” applications
- Optima
  - System and application debug tool suite built on Eclipse
- Netbricks
  - Protocol stacks for TDM, 3G, IMS, and NGN networks
- Polyhedra
  - A high-performance, low-footprint relational database system designed for embedded use
- Element Device Manager
  - Control the capabilities of dSPEED by loading and booting cores with appropriate run-time images, monitoring the state of processor cores, and restarting them after fault conditions

OSE 实时操作系统版本

- **OSEck for DSPs (10 kB+)**

  专门为 **DSPs**优化设计的 **RTOS**

- **OSE 5.x for CPU 结构 (100kB+),例如ARM,PPC,MIPS**

  功能丰富的嵌入式实时操作系统 **(RTOS)** 平台

- **OSE Epsilon for MCU/SoC (4kB)**

  性能高、代码小的 **RTOS (100%** 汇编编写的内核**)**

OSEck支持绝大部分的 DSP 和 CPU

- TI's TMS320C54x
- TI's TMS320C55x
- TI's TMS320C62x
- TI's TMS320C67x
- TI's TMS320C64x
- TI's TMS320C64x+ (6482/3/4, 6486/7/8...)
- TI Shannon/ Nyquist (going)
- Freescale MSC8101/03
- Freescale MSC8102/22/26
- Freescale MSC8144
- Freescale MSC815x(beta)
- CEVA-X and CEVA-TeakLite-III
- NIOS II
- SC120/SC1200
- SC140/SC1400
- SC2400/SC3400
- ADI TigerSHARC (TS101S and TS201S)
- ADI Blackfin
- Freescale MPC5xx/55xx
- LSI Logics ZSP400/500
- STMicroelectronics ST100/120/…
- Agere Systems dsp16k
- ARM
- …..

OSE 5.x 支持大部分的 RISC CPU

- Freescale PowerPC family: 5xx, 6xx (603, 603e, 603ev, 604, 604e), 7xx (740, 745, 750, 755), 74xx, MPC 8xx, MPC82xx PQII, MPC83xx (8347/8349), MPC 85xx (854x, 856x) PQIII, MPC 8641D
- ARM consortium family: ARM 7, ARM 9, ARM10, ARM11, XScale
- AMCC PowerPC family: 4XX (403, 405EP, 405GP, 440GP, 440GX), NPe405H, NPe405L, PowerNP4GS3)
- IBM PPC 7XX (750, 750CX, 750FX, 750GX),
- Intel Network Processor: IXP2400, IXP2350..
- Intel Strong ARM family: StrongARM, XScale incl. PXA25x
- Texas Instruments: OMAP (all chipsets, for OMAP DSP part see OSEck)
- MIPS 32/64 consortium family
- Cavium MIPS based Multi-core

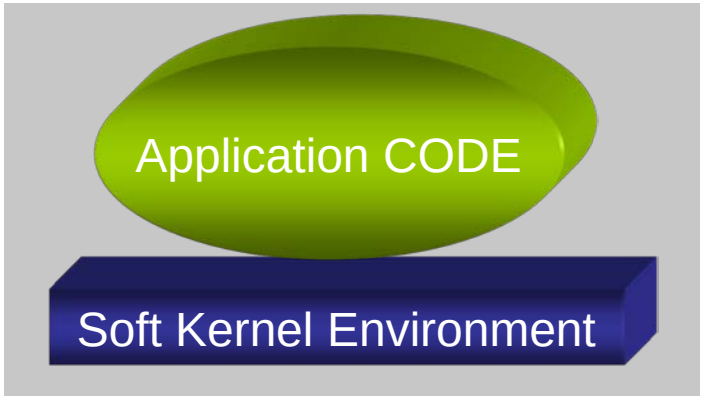
OSE RTOS体系构架

- 针对**RISC, DSPs** 和 **MCUs**有三种不同的优化实现
  - 提供一致的基本API
  - 为每个硬件结构高度优化的RTOS
  - 基于相同的开发工具：软内核(Soft Kernel) 和 Illuminator
- 直接消息传递**(Direct Message Passing)**结构
  - 简化系统设计
  - 容易学习和使用
  - 代码简洁，可重用性高
  - 支持从最简单的系统，到复杂的多核、多**CPU/DSP**、多板系统
- 集中式的出错处理机制
- 完全可抢占式和可确定性的硬实时内核

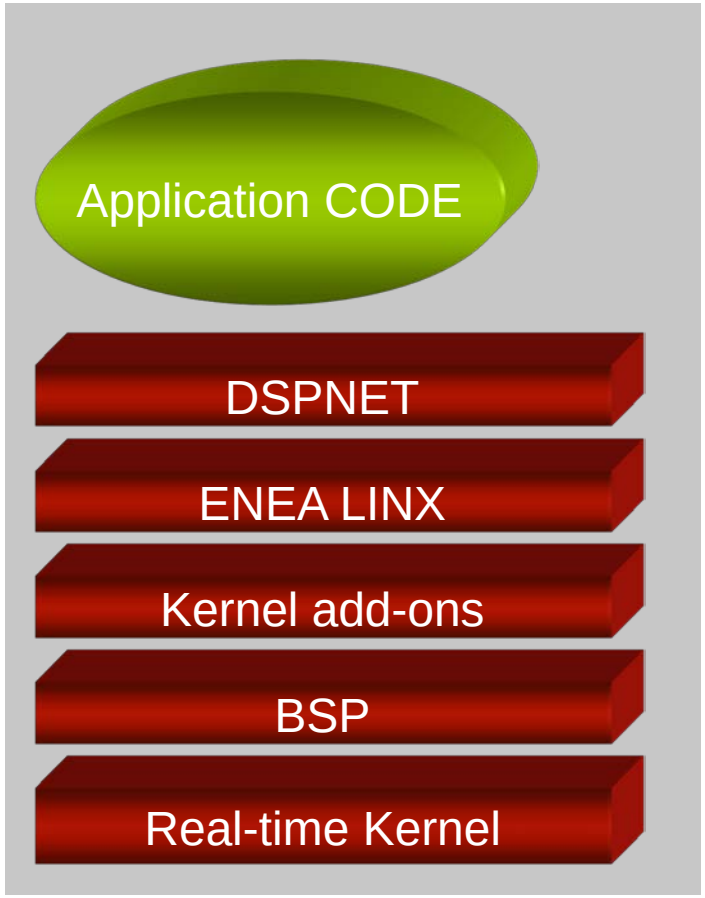
OSEck Components

Host: Tools

Host: SFK Simulator



TARGET

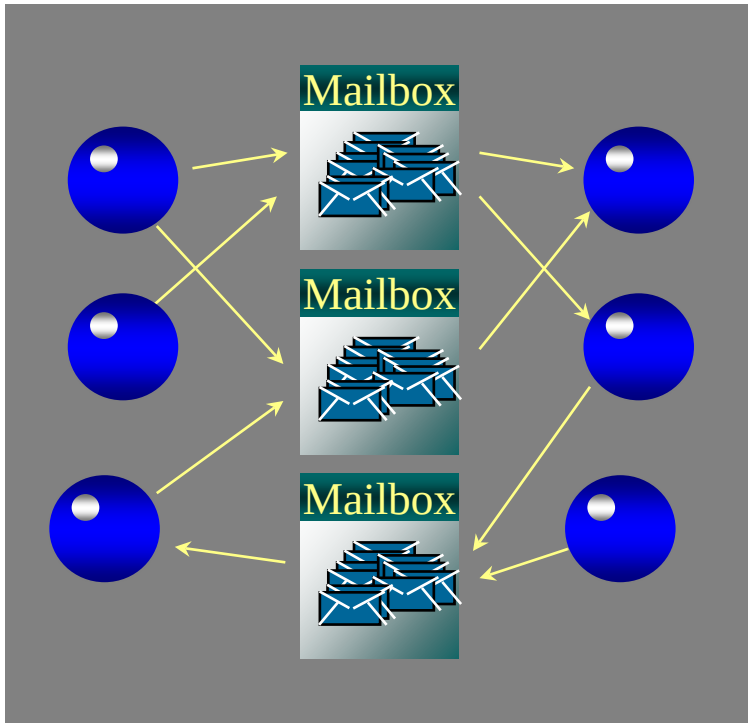


OSEck Kernel

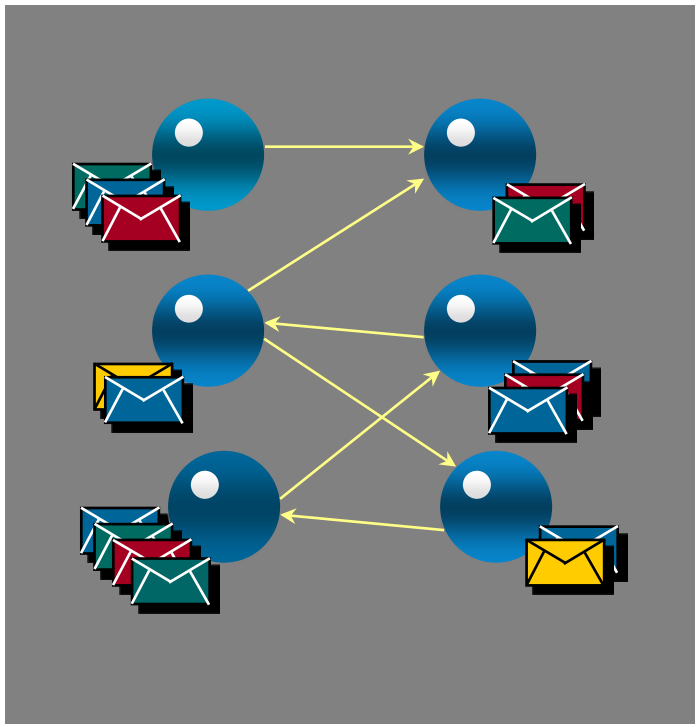
- Fully pre-emptive and deterministic kernel
- Scalable based on use of system calls
- Low interrupt latency and high data throughput
- Dynamic creation and termination of processes
- Dynamic memory allocation with bounded execution time
- No memory fragmentation
- Built in error management
- Power management
- Built in debug and trace functionality

The New Generation RTOS

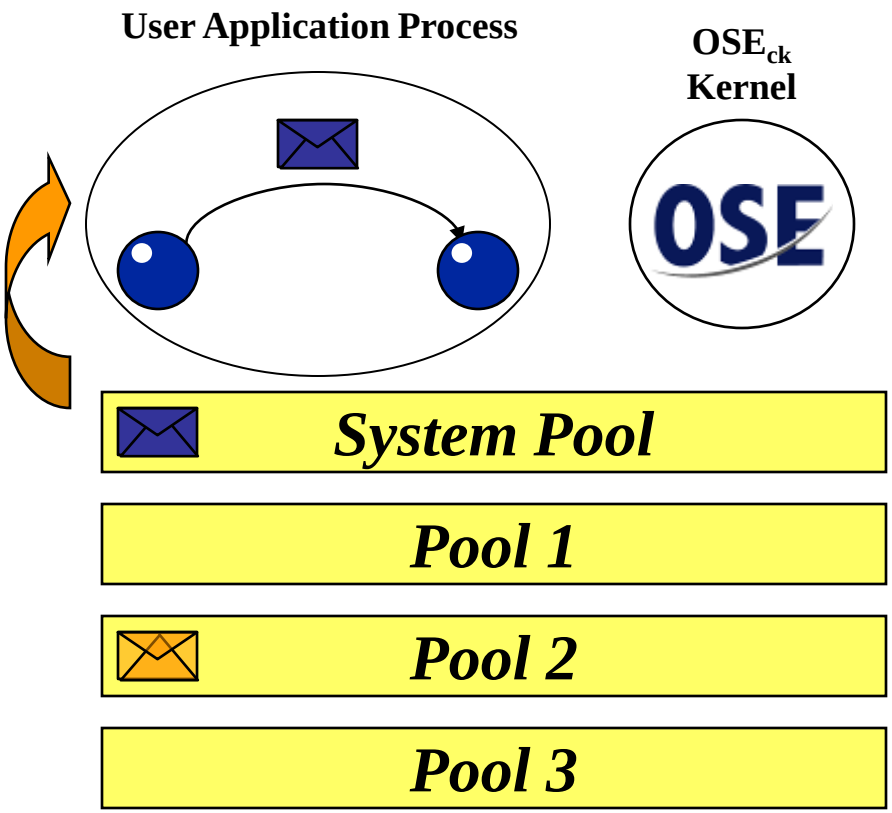
Conventional mailbox



OSE Message

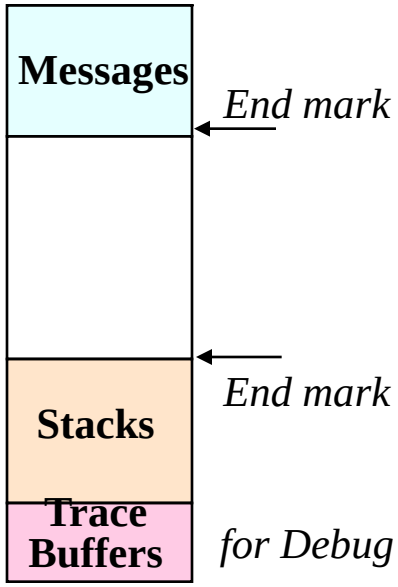


Memory Pools

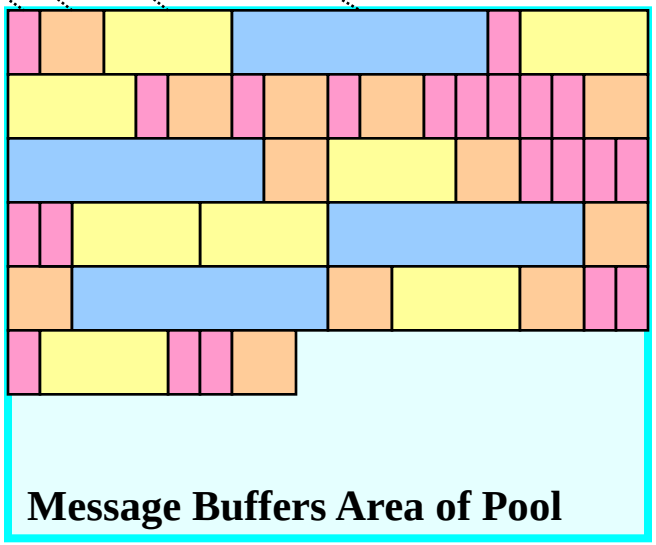


- Pool is the area of memory used for Allocation of …
  - Signal Buffers
  - Stacks
  - Kernel Memory
- Memory is allocated and liberated **dynamically**
- When using ***alloc()*** the buffer is allocated from the system pool
- When using ***s_alloc()*** the buffer is allocated from a specific memory pool

没有碎片的内存管理

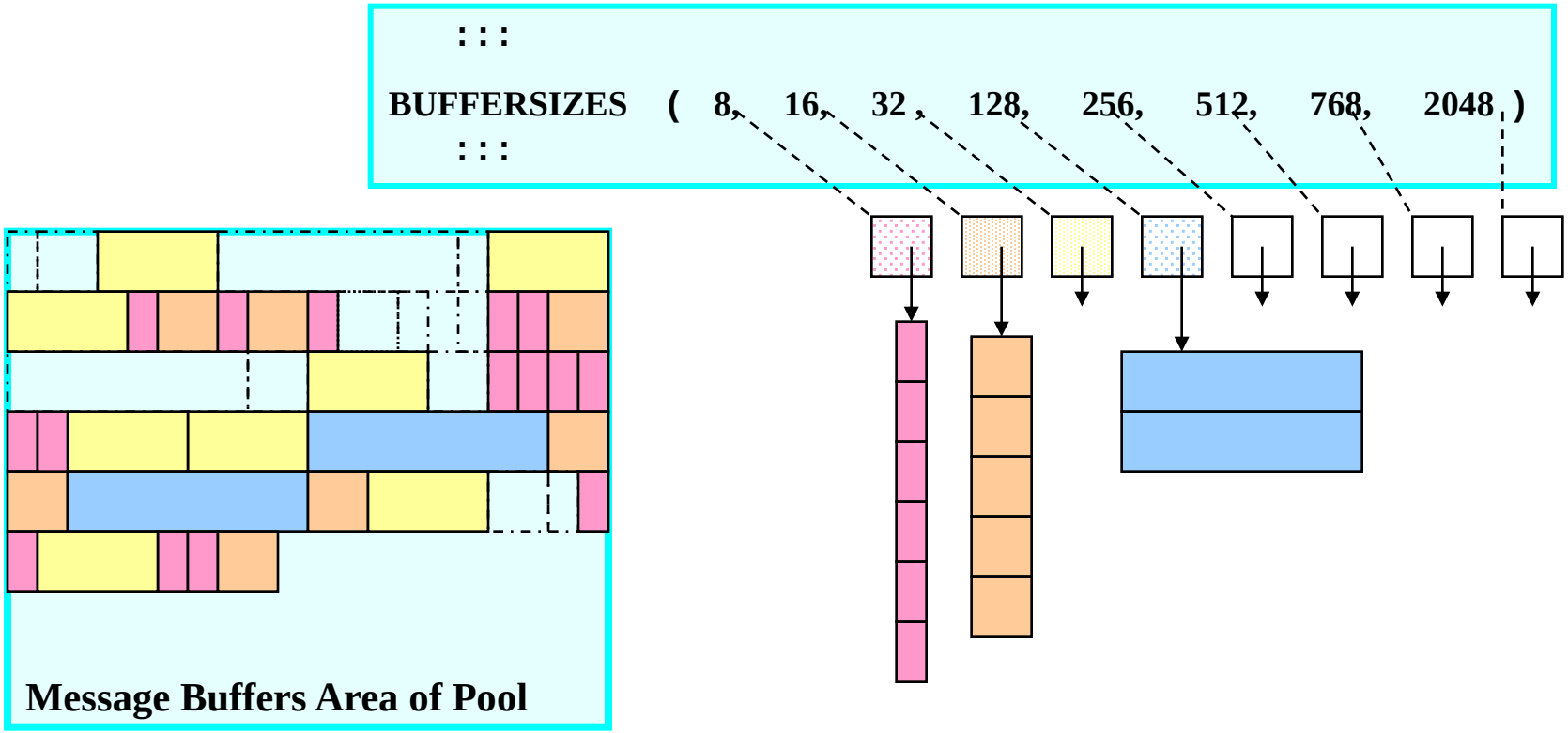



```
   :::
BUFFERSIZES( 8,  16,  32,  128,  256,  512,  768,  2048 )
   :::
```



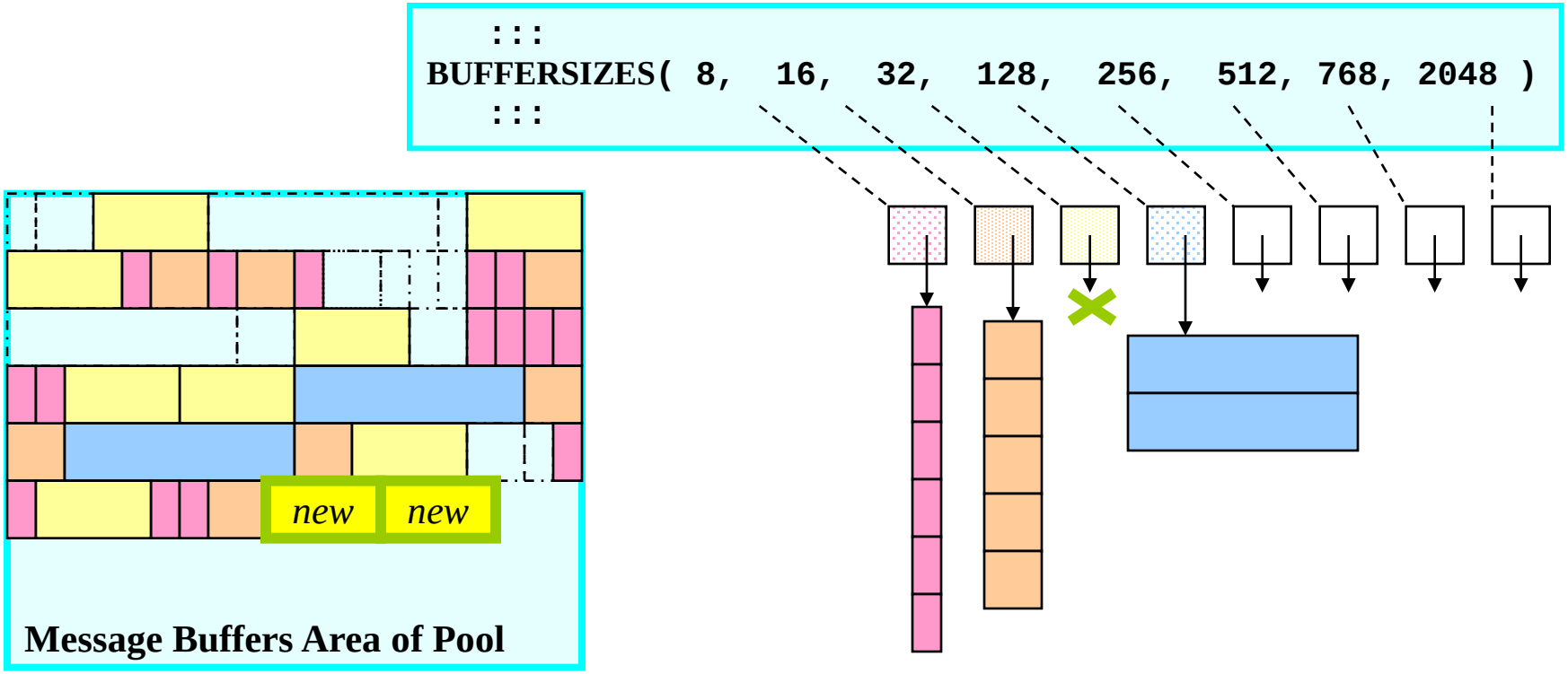
- **Benefits:**
  - **Very high performance**
  - **Deterministic allocation time (always)**
  - **Cache optimized**
  - **No fragmentation**

重新使用缓冲



- The Free Lists
  → Recycling at Same Size

没有空闲已分配内存时

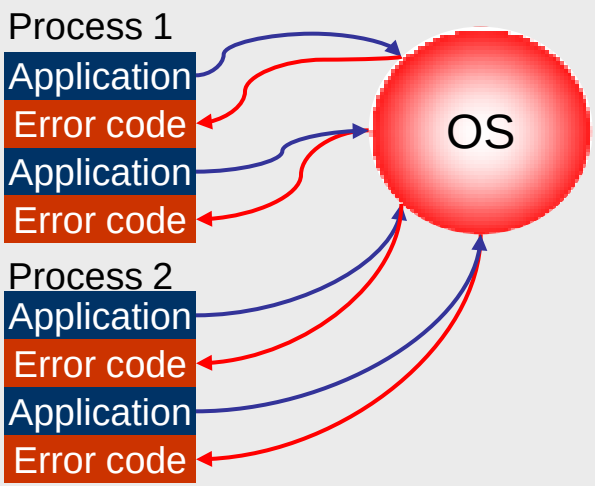


- Buffers Area of Pool is Enlarged

OSEck Error Handler

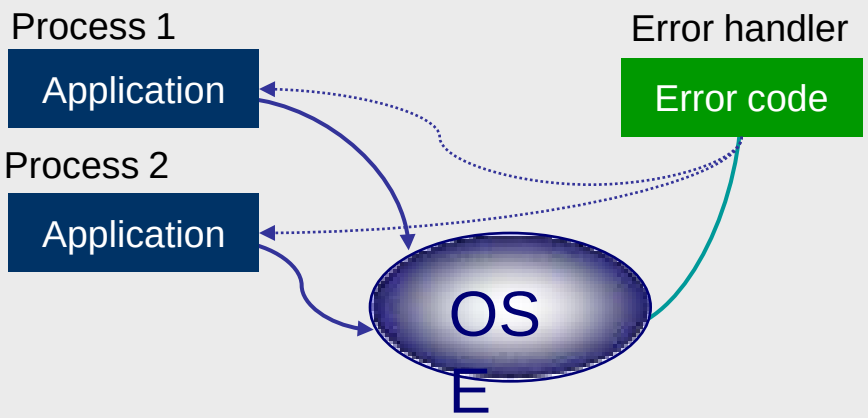
Error detection and handling is done by the Kernel

Conventional RTOS



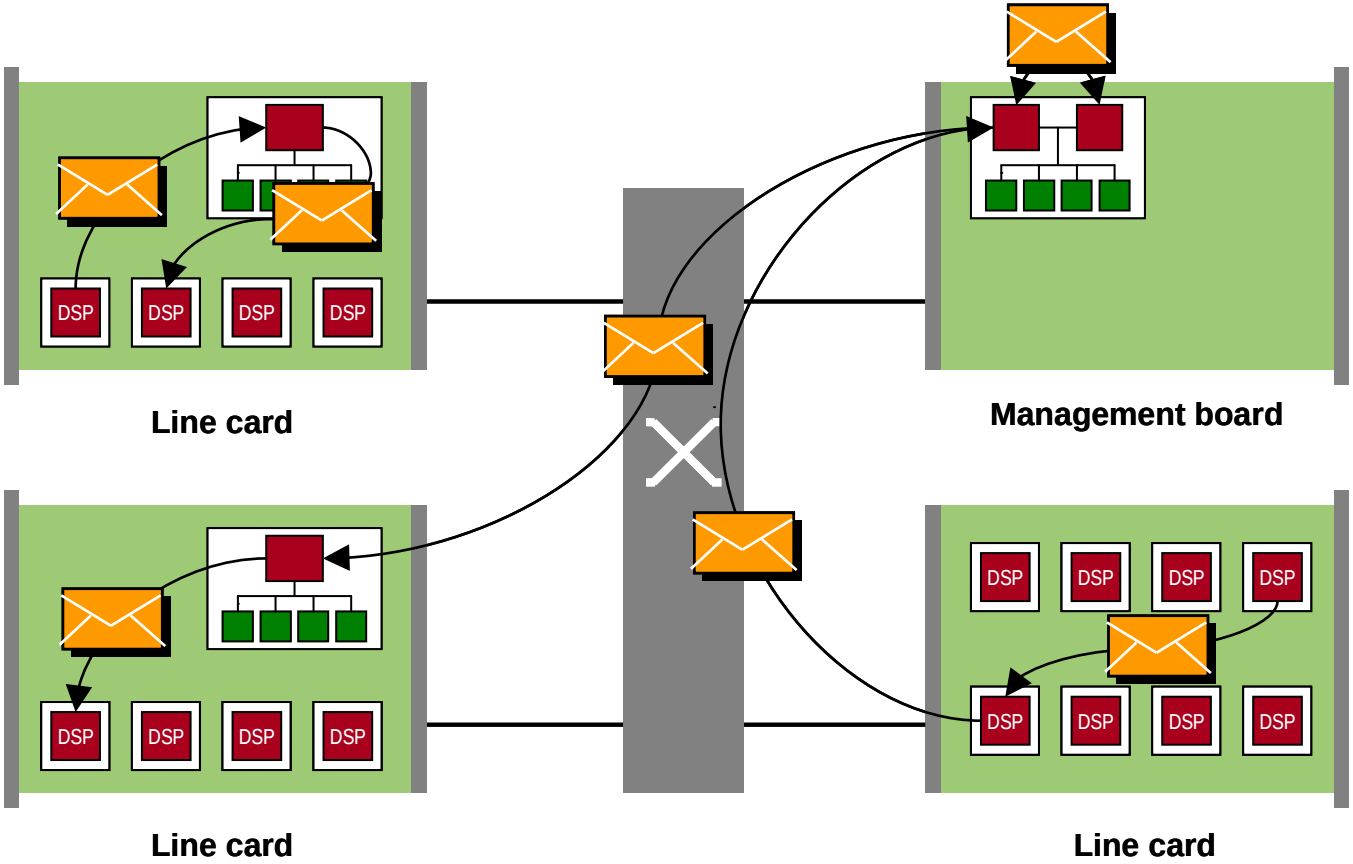
Errors are managed within the process by the process code

OSE Error Handling



In case of an Error detected by the kernel It will automatically go to the Error Handler

LINX extends Message Passing to Boards



OSEck for DSP Multi-Core

- Multicore support
  - Shared code
    - Saves memory since all cores runs the same code
  - Shared dynamic memory/data
    - Memory can be allocated by one core and freed by another
  - Zero copy “message passing” between cores.
    - Same IPC mechanism as in one core (Transparent to the user)
    - Data can be transferred between cores without being copied

- Memory protection
  - Error pin pointing
  - ….

OSEck Core Dump Analyzer (CDA)

- **CDA is an ”off-line” utility that analyzes the crash dump**
  - Error Messages – identify error type, location, and process ID
  - All Process Lists/States
  - Individual Process Information
  - Signal Queues
  - Memory Pools Summary – stack, signal buffers, allocated, free
  - Signal buffer analysis – finds corrupted signals
  - Summary of all signal buffers in system
  - All system and process stack and register settings

Optima System Browser

Hierarchical view
(System model)
(Context menus)

Gateway

Target

Block

Processes

Type and state
decorations

Load modules
view (Context
menu)

Editor
navigation
Back / Forward

Details editors
(System model)
(Double click)

Table data views
(Block)
(Process)

On target filter
(All properties)

Sortable
(All properties)

Resource - ose_monitor (0x10007) - Eclipse Platform

File Edit Refactor Navigate Search Project Run Window Help

Resource

OSE System Browser

sfk-win32 172.16.140.77:21768
sfk-win32 172.16.140.7:21768
sfk-linux 172.16.230.20:21768
172.16.230.20:21768
pool (0x10000)
main (0x10008)
OSE (0x10001)
ose_monitor (0x10007)
ose_huntd (0x10006)
ose_huntd (0x10005)
ose_tickd (0x10004)
ose_sysd (0x10003)
idle (0x10002)

main (0x10008) | ose_monitor (0x10007)

ose_monitor

Process Information | Signal Select | Signal Queue

Killed:	No
Name:	ose_monitor
Process ID:	0x10007
Block ID:	0x10001
User Number:	0
Type:	Prioritized
State:	Running
Priority:	1

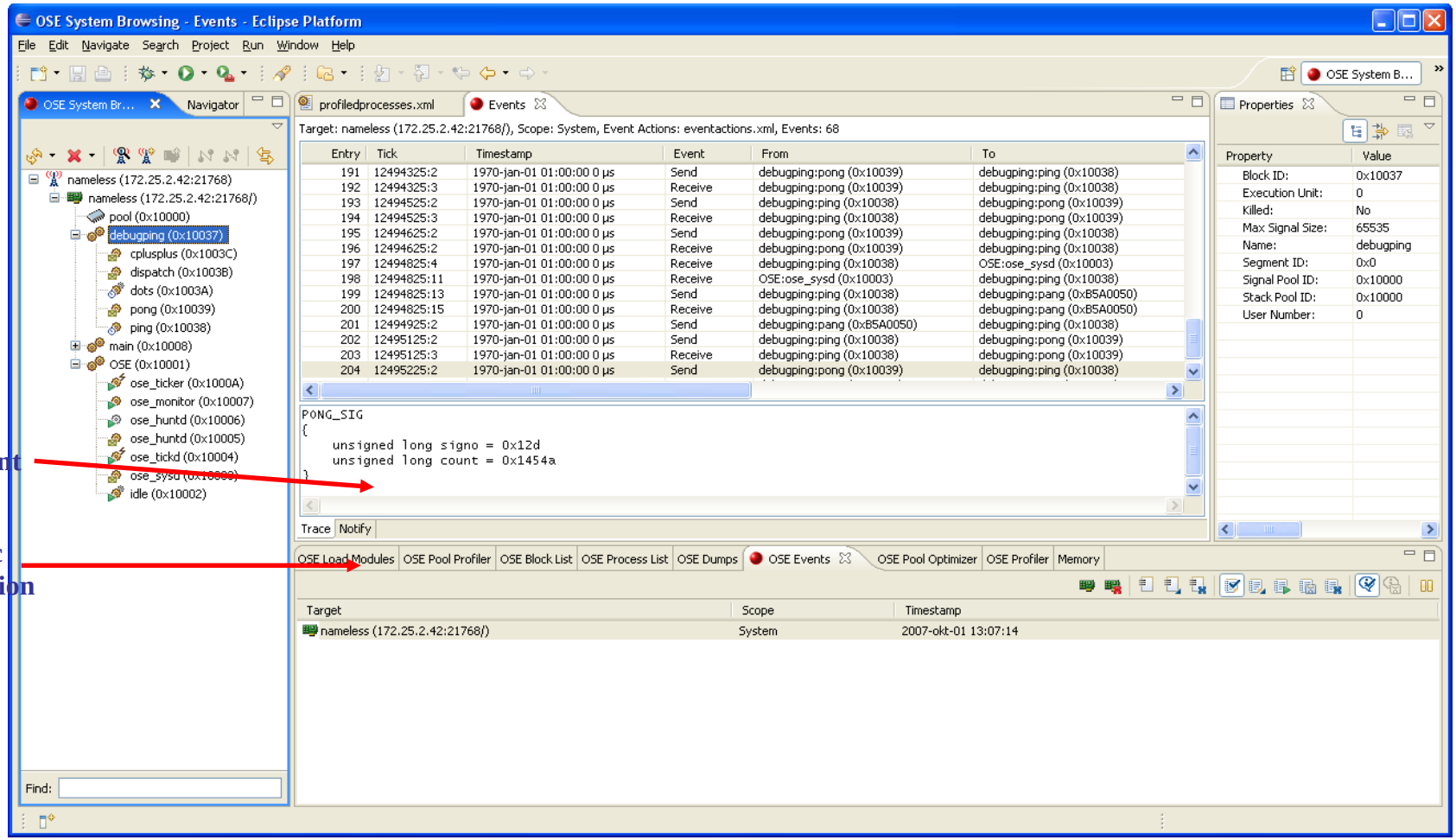
Value	Sig No

OSE Load Modules | OSE Block List | OSE Process List

Target: 172.16.230.20:21768, Processes: 45

Name	PID	BID	User	Type	State	Priority	Sigs in Q
FAM_COMMON_SENDER	0x10019	0x10008	0	Phantom	Ready	0	
core_supervisor	0x1000B	0x10008	0	Prioritized	Receive	15	
echo	0x1002F	0x10008	0	Prioritized	Receive	7	
main	0x10009	0x10008	0	Prioritized	Receive	16	
netw_supervisor	0x10023	0x10008	0	Prioritized	Receive	15	

Optima System Events



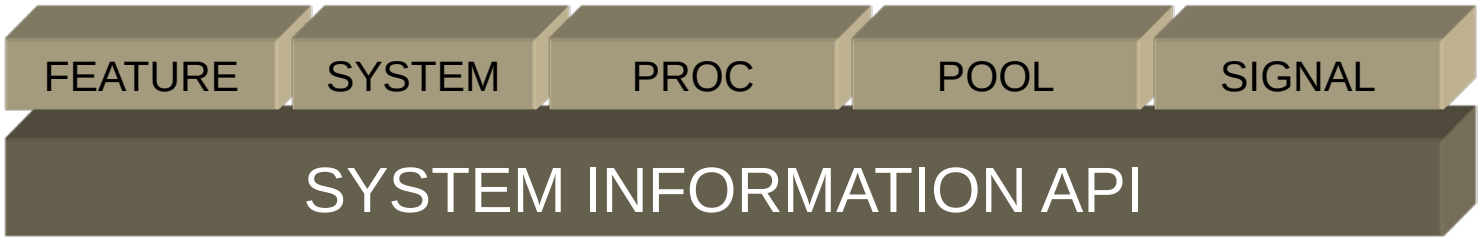
Send event

Symbolic information

Optima OSE Memory Pool Profiler

OSEck System Information Interface

- Methods of retrieving information from an OSEck System not available from the standard system calls.
- Can be used for run-time debugging, profiling and tuning of the system.
- Generic interface and one extension interface. The extension interface is unique for each target.
- The interface is divided in five groups for different inquiries:



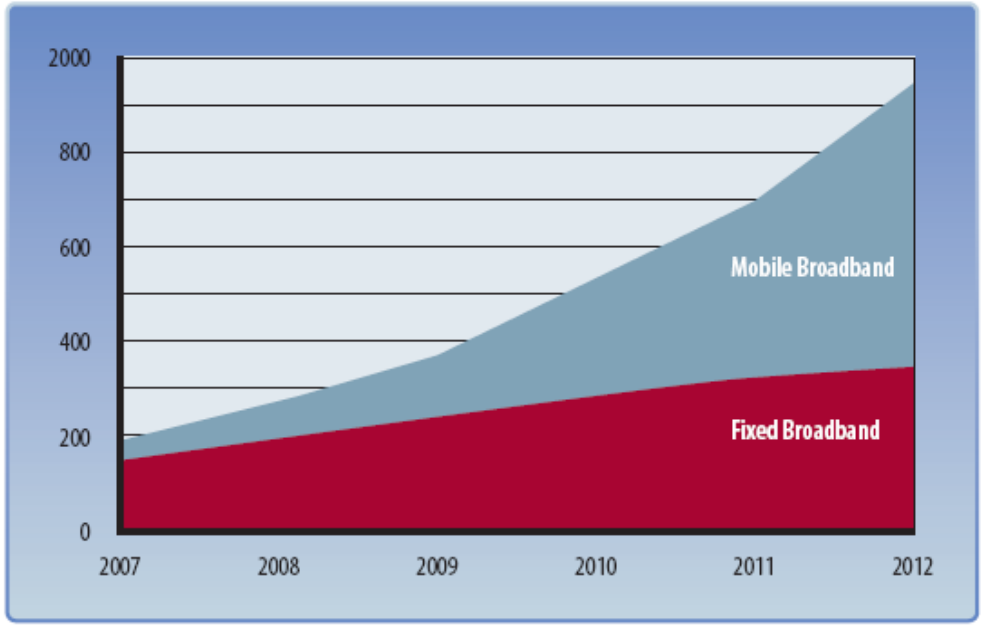
OSEck RTOS Benefits

- Customer proven, Stability – 20+ years filed verification
- Cross platform - support almost all the DSP with same APIs.
- High performance, small code size, Scalable
- LINX - Transparent IPC for distributed System, support lots of medias.
  - Core to Core;
  - DSP to DSP/CPU
  - Board to Board
- Good Error process mechanism
- Good memory management mechanism.
  - High Performance, Deterministic
  - No Memory fragmentation
- Good tools,
  - Core Dump Analyzer,
  - Optima – Eclipse based tools.
  - Soft Kernel – PC simulation tool
- Easy coding, easy learn. Short Development time
- Fault Tolerance, Failure Detection.
- Power management
- Best Local support
- ….

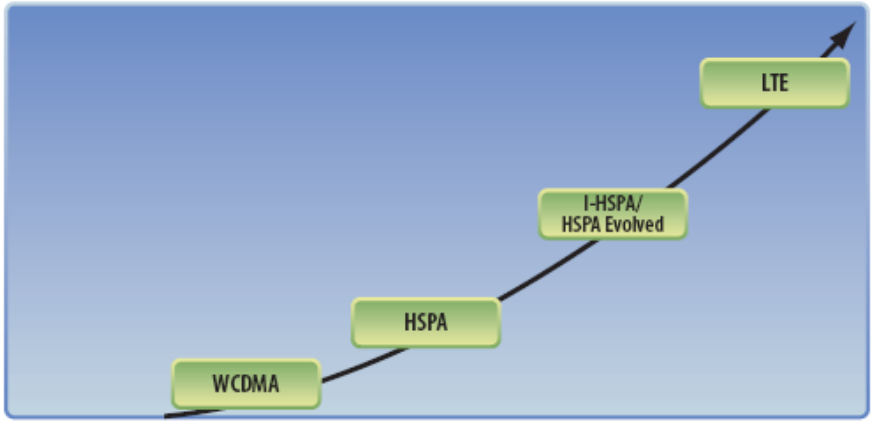
Agenda

- Enea 简介
- Dataplane Processing Platform
  - OSEck RTOS for DSP
  - dSPEED – DSP 集群软件管理平台
- 总结

Exponential Data Traffic Growth



Broadband subscriber growth



Peak Data Rate	0.384 Mbps DL 0.64 Mbps UL	14 Mbps DL 6 Mbps UL	28/42 Mbps DL 12 Mbps UL	160 Mbps DL 50 Mbps UL
Available from Equipment Manufacturer	2001	2006	2008	2009

Data Bandwidth growth

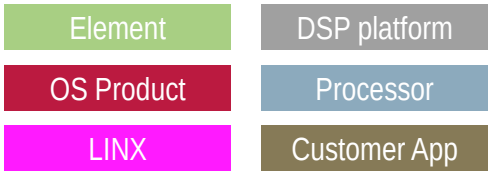
Clustered DSPs Address these Trends, but...

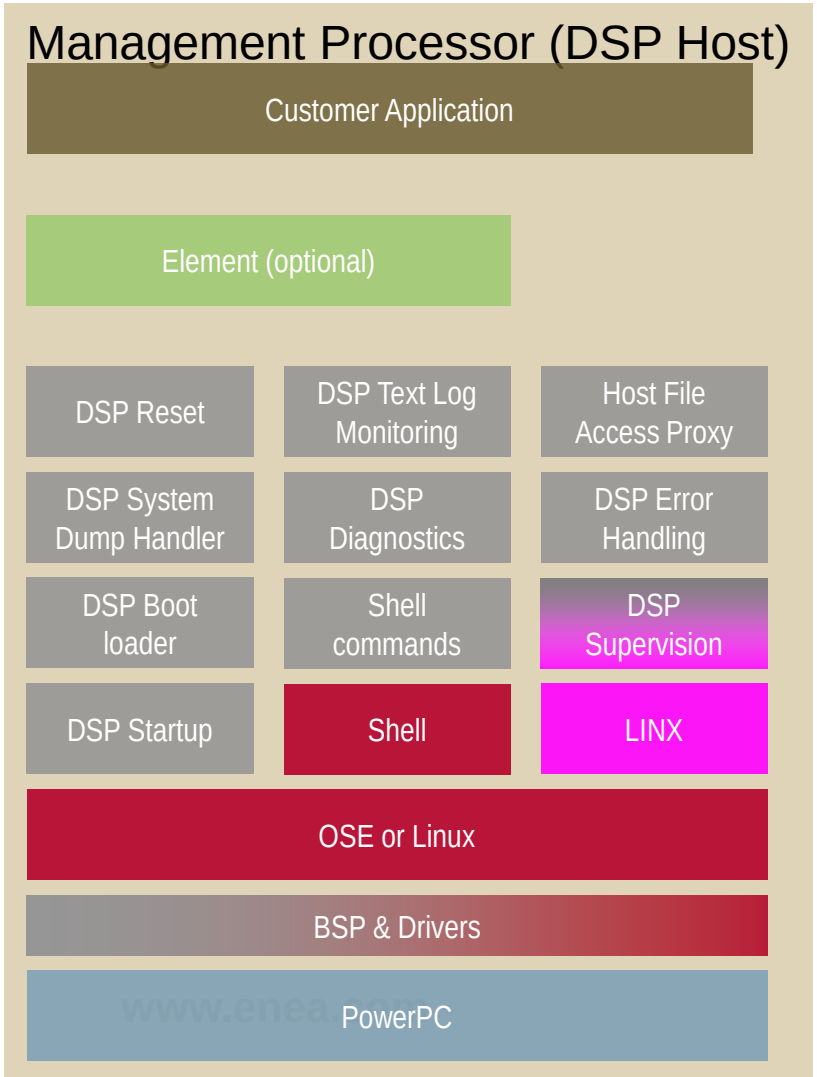
How do you:

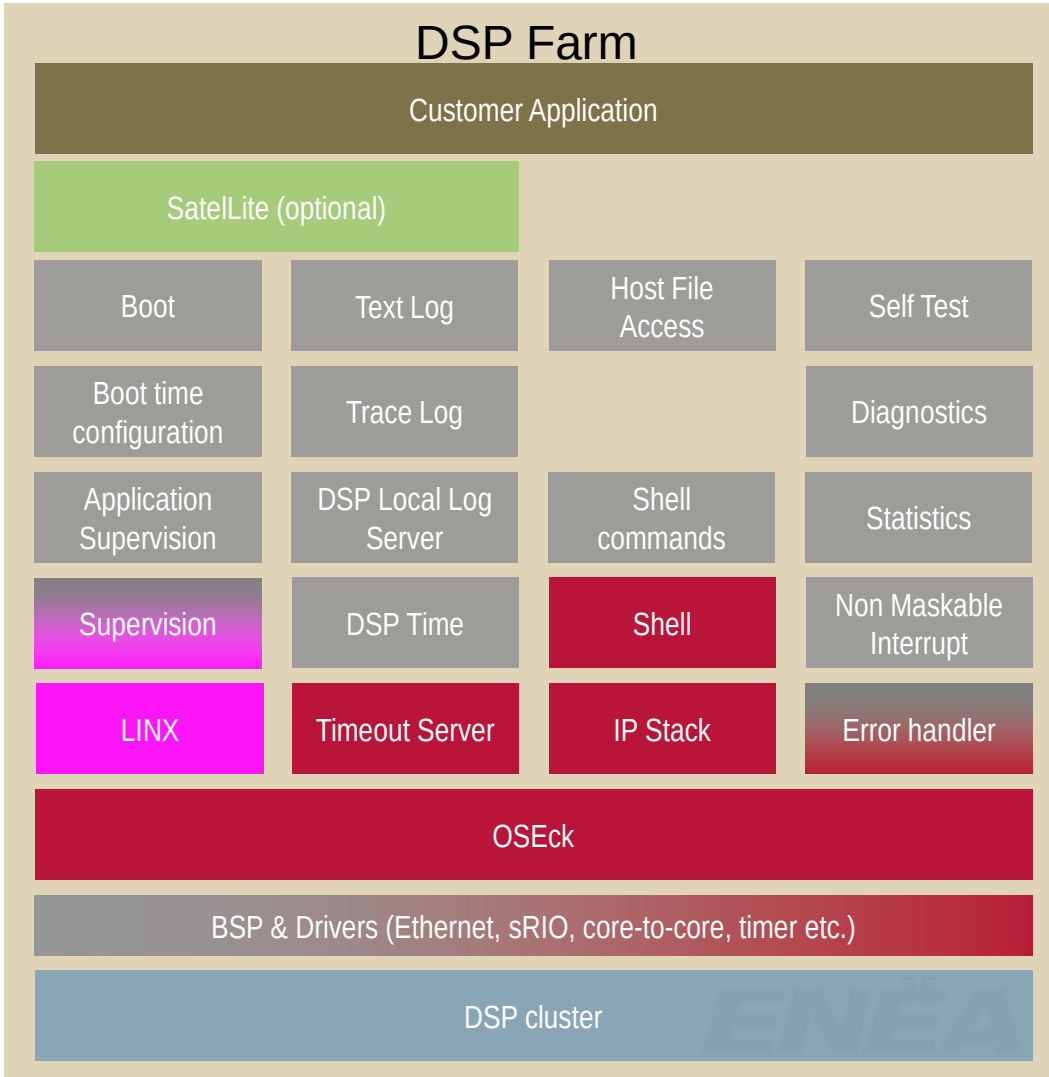
- Manage these multi chip, multi core DSP compute environments.
- Set up and configure a cluster of (multi-core) DSP's
- Debug multiple DSP-farms from a system wide perspective with 4,8,16+
- Handle and manage errors to preserve robustness in high channel density systems and multicore devices ensuring 99.999% uptime
- Find the root cause of a problem when debugging distributed heterogeneous deployed systems
- Ensure maximum system performance

The Answer

dSPEED - DSP SW Platform







www.enea.com

dSPEED Services

DSP Boot and Reset

- Reset / Release
- Flexible Boot loading of the DSP's. (sRIO, Ethernet)
- Start Up (Initialize system, Run self tests on HOST and DSP, Start Platform and applications separately)

Error Detection and Handling

- Processor, kernel, user error
- Isolate failing DSP cores and put the DSP in a controlled state

Supervision

- Basic link level supervision mechanism provided by Enea LINX
- Platform level supervision mechanism to detect overloaded DSPs and out of control DSP crash
- DSP applications can subscribe to supervision signals
- DSP watchdog can optionally be used as an additional supervision mechanism

Logging

- DSP Log (Controls and configures the log policies of the Text and Trace Log)
- Text Log
- Trace Log

Core Dump

- Normally generated at program or processor crash
- Produce a report that contains comprehensive analysis in order to reveal different kinds of errors
- Gives static system information
- Core Dump Analyzer (CDA) Tool
- Embedded Core Dump Analyzer (ECDA) Library
- Text, trace log and user defined raw memory areas can optionally be included in the dump
- Dump over RapidIO or Ethernet

Shell and Shell Commands

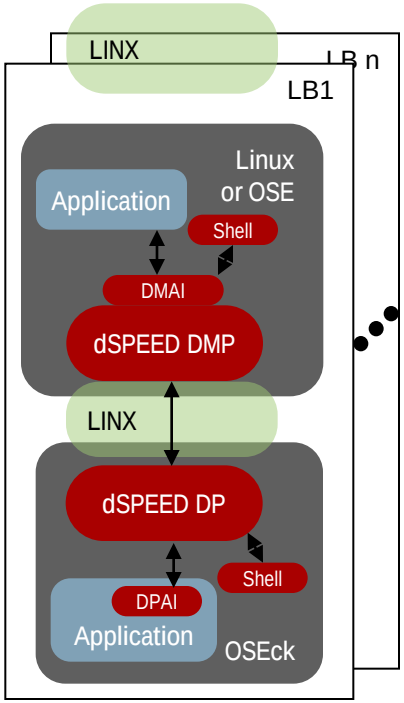
- Connectivity via Ethernet (telnet), serial etc. to any CPU
- Shell Commands
- CLI to all relevant APIs in the platform

Statistics

- Monitoring of OS and User level statistics
- Driver level statistics
- Protocol level statistics
- User level statistics

Host File System Access

- Open, create, write, read, close, seek or get statistics of the files in the file system on the host



DSP Reset and Boot Architecture

- Reset / Release
  - The host application control individual DSP reset and release
  - Adaption layer to specialized hardware support (e.g. CPLD) or sRIO reset
- Flexible Boot loading of the DSP's. Two boot boot modes supported (at least)
  - sRIO
    - The host load code and data directly into the DSP memory in sRIO boot mode.
  - Ethernet
    - In Ethernet boot mode a 1st stage bootloader (usually I2C to an EEPROM) is used to load a minimal second stage bootloader.
    - Sometimes a 1st stage
  - Start up
  - Initialize system
  - Run self tests on DSP
  - Start platform and applications separately
  - Supervision of DSP's is started when DSP's have setup a LINX connection to the host.

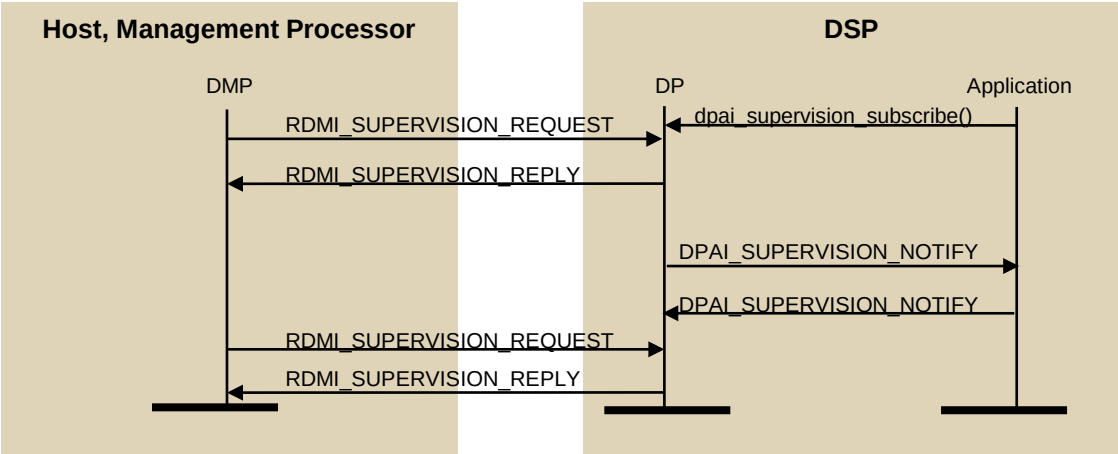
Boot Time Configuration

- Key-value file downloaded to DSP together with DSP boot image.
  - E.g. Host and DSP MAC adresses.
  - DP application specific configuration.

- Symbol information stored by host in hash table to access correct areas in memory. No hard coded areas necessary.
  - Fatal error information struct
  - Logg buffers
  - Other variables / structs of interest. Accessible e.g. after a crash

Supervision

- Basic link level supervision mechanism provided by LINX
- Platform level supervision mechanism to detect overloaded DSPs and out of control DSP crash.
  - NMI is used to take control over a potentially overloaded and free running DSP
- DSP applications can subscribe to supervision signals.
  - If a DSP application does not respond, the DMP is notified by the DP.
- DSP watchdog can optionally be used as an additional supervision mechanism
  - Typically generates a NMI



Error Detection and Reporting

- Processor error
  - Memory access exceptions, illegal instruction etc.
- Kernel error (OSEck software error)
  - Errors detected by the kernel, bad parameter to system call, memory corruption etc.
- User error
  - Any application error
  - Reported as fatal or non fatal through DPAI

Fatal Error Handling – Direct Memory Access

- Direct memory access available. E.g. RapidIO or HPI
  - Requires only communication peripheral to be able to execute
- A fault message will be entered into memory accessable by host
  - Locked interrupts, minimal functionallity used.
  - The error informaiton contain
    - CoreID
    - Error code + additional information
    - File and line
    - Error message string
- Interrupt line to host can be triggered to indicate that host should look for fatal error information
- NMI can be used to set the DSP core in a safe state (fatal error state) which enable other DSP cores to finish an ongoing task before a restart.

Fatal Error Handling – No Direct Memory Access

- No direct memory access. E.g. RapidIO or HPI
  - Requiers CPU to be able to execute
- A DP user defined callback will be called indicating
  - Error code
  - Additional information
  - Error message string
  - File and line information
- If an Ethernet connection available an error frame is broadcasted
  - Locked interrupts, minimal functionallity used.
  - The error frame contain:
    - CoreID
    - Error code + additional information
    - File and line
    - Error message string
- Interrupt line to host can be triggered to indicate that host should look for fatal error information
- NMI can be used to set the DSP core in a safe state (fatal error state) which enable other DSP cores to finish an ongoing task before a restart

Post Mortem Debugging

- System dump normally generated at program or processor crash
  - DSP indicates a fatal error indication through DP error handler
  - DMP -> DP supervision timeout
    - NMI is used to take control over a potentially overloaded and free running DSP to avoid the DSP ECDA to fail.
- Contains lots of information in order to reveal different kinds of errors
- Gives static system information

```
3. Signal Queues
 ================

 ================================================================================
 Process: Pid = 0 (myIdle)
 The signal queue contains no signal buffers.

 ================================================================================
 Process: Pid = 1 (timer0_int)
 The signal queue contains no signal buffers.

 ================================================================================
 Process: Pid = 2 (master)
 The signal queue contains 9 correct signal buffers.

 Signo Send Addr Size   Pool Hex Addr  0000 0001 0002 0003 0004 0005 0006 0007
 ----- ---- ---- ------ ---- --------  ---- ---- ---- ---- ---- ---- ---- ----
     4    1    2    500    0     286e  0004 dc06 0000 0000 0000 0000 0002 d8c4
     4    1    2   1000    1     859a  0004 0072 0011 f7bb 00ca fff8 0100 fe29
```

dSPEED System Dump Information

DP Information

- Text log
- Trace log
- Statistics
- User defined memory sections
- DSP app defined memory sections

Error messages

Processes

Signal queues

Memory usage

Signal space usage

Signal space analysis

Listing of all signals in the system

Saved registers and stack content

DP specific system information

Only available in system dumps generated from DMP

Standard core dump containing OSEck system information

Available in status reports generated by CDA and ECDA

Statistics

- **Monitoring of OS level statistics**
  - CPU/Process load
  - Memory consumption
  - User defined
  - High/Low Watermark info
- **Driver level statistics**
  - Ethernet
  - sRIO
  - Shared memory
- **Protocol level statistics**
  - IP
  - LINX
- **User statistics**
  - Users can register their own statistics with the statistics framework
- **Statistics can be retreived from all interfaces. DMAI, DPAI and CLI.**
- **Statistics can be monitored by a Host application, DSP application or shell.**

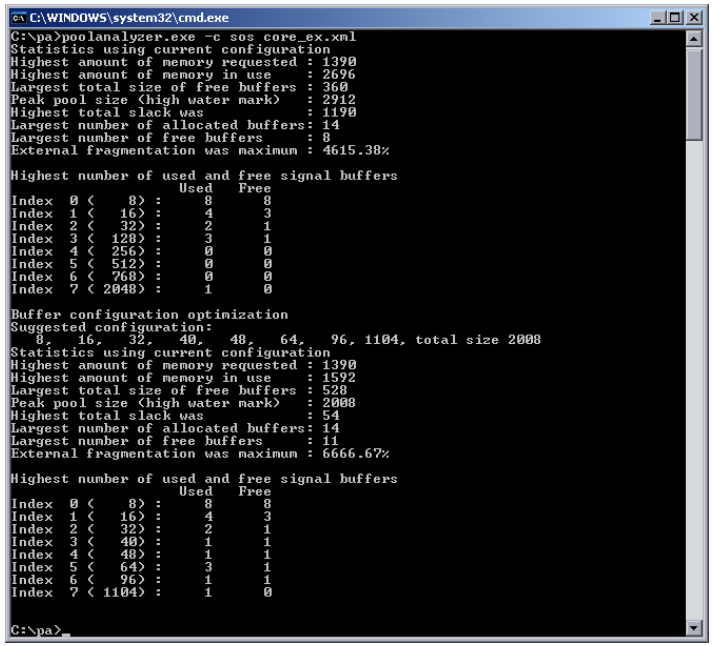
DSP Diagnostics

- The purpose of Diagnostic tests is to verify that important HW functions are working correctly

- Three levels of diagnostics:
  - Offline diagnostics
    - Destructive tests can be uploaded to the DSP
  - Self test is short version to test vital part necessary for successful booting
    - Memory, core and Ethernet and sRIO
  - A long version provide a full test suite
    - Memory, peripherals, clock, timer and CPU/DSP core test

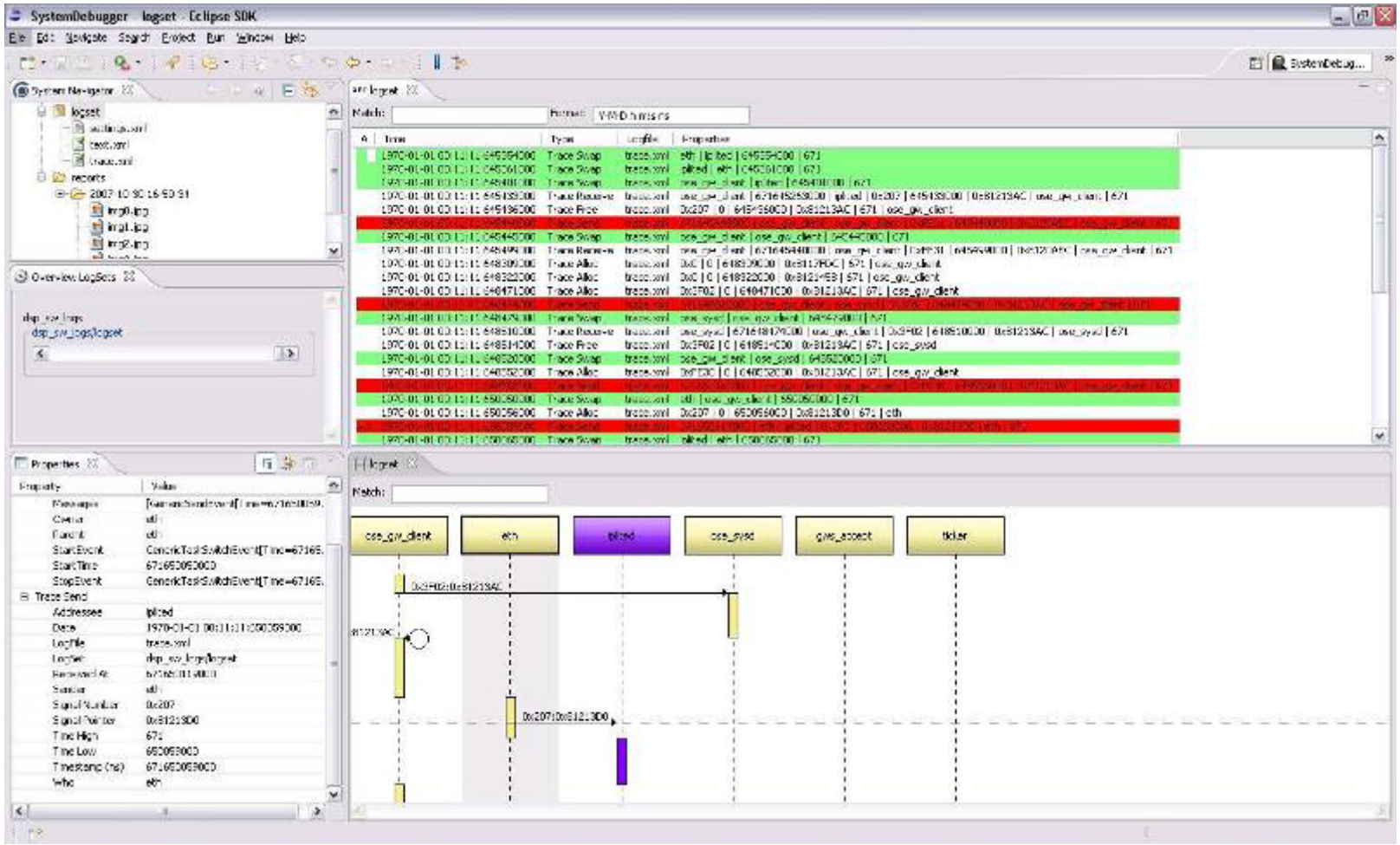
Shell and Shell Commands

- OSEck shell
  - Easy to connect to using telnet when used toghether with DSPNet (IP stack)
  - LINX, serial etc,
- DMP CLI available from Linux or OSE shell
- Very useful for debugging purposes.
- Many Commands for platform functionality available
  - Text & Trace log
  - Statistics
  - Etc.
- CLI to all relevant APIs in the platform



```
C:\WINDOWS\system32\cmd.exe
C:\pa>poolanalyzer.exe -c sos core_ex.xml
Statistics using current configuration
Highest amount of memory requested : 1390
Highest amount of memory in use    : 2696
Largest total size of free buffers : 360
Peak pool size (high water mark)   : 2912
Highest total slack was            : 1190
Largest number of allocated buffers: 14
Largest number of free buffers     : 8
External fragmentation was maximum : 4615.38%

Highest number of used and free signal buffers
                        Used   Free
Index  0 (     8) :      8      8
Index  1 (    16) :      4      3
Index  2 (    32) :      2      1
Index  3 (   128) :      3      1
Index  4 (   256) :      0      0
Index  5 (   512) :      0      0
Index  6 (   768) :      0      0
Index  7 ( 2048) :      1      0

Buffer configuration optimization
Suggested configuration:
    8,   16,   32,   40,   48,   64,   96, 1104, total size 2008
Statistics using current configuration
Highest amount of memory requested : 1390
Highest amount of memory in use    : 1592
Largest total size of free buffers : 528
Peak pool size (high water mark)   : 2008
Highest total slack was            : 54
Largest number of allocated buffers: 14
Largest number of free buffers     : 11
External fragmentation was maximum : 6666.67%

Highest number of used and free signal buffers
                        Used   Free
Index  0 (     8) :      8      8
Index  1 (    16) :      4      3
Index  2 (    32) :      2      1
Index  3 (    40) :      1      1
Index  4 (    48) :      1      1
Index  5 (    64) :      3      1
Index  6 (    96) :      1      1
Index  7 ( 1104) :      1      0

C:\pa>_
```

dSPEED Trace Log Visualization in Optima



Agenda

- Enea 简介
- Dataplane Processing Platform
  - OSEck RTOS for DSP
  - dSPEED – DSP 集群软件管理平台
- 总结

Enea Software in Products

- **2/3 G, HSPA, LTE Infrastructure**
  - Radio Base Stations
  - Radio Network Controller
  - Gateways
  - Core Network
  - …..
- **2/3G Handheld Devices**
  - Mobile phones
  - Tetra Terminals
- **Software Radio, IPTV**
- **Automotive**
  - Engine Control
  - Telematics
- **Military**
  - Radar / Sonar
  - Communication Equipments
  - Warship
  - …..

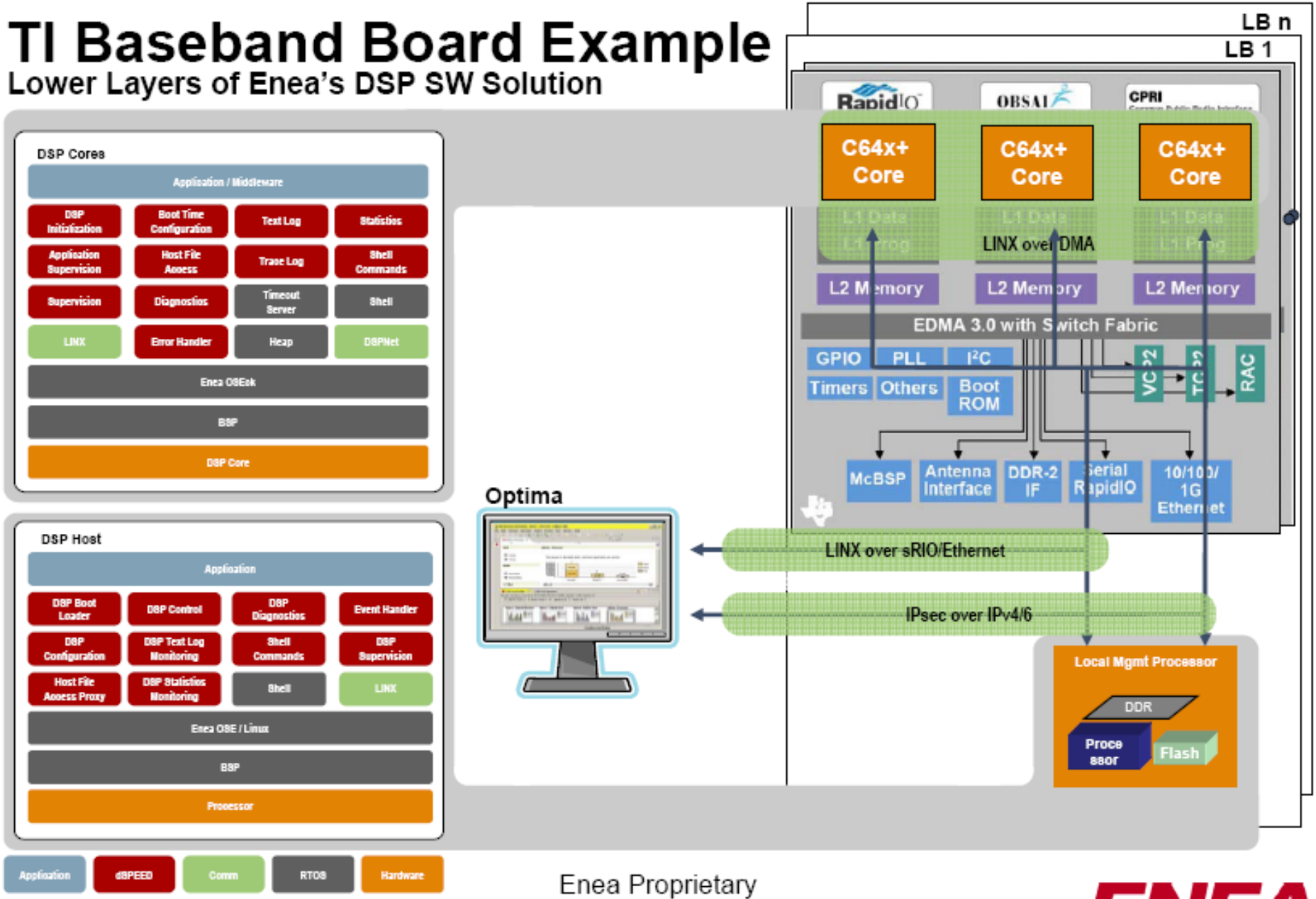
Product Characteristics

- Data / Control centric
- Distributed Systems
  - Cluster solution
  - Multicore
- Heterogeneous
- High Performance Requirements

dSPEED Demo

TI Baseband Board Example

Lower Layers of Enea's DSP SW Solution

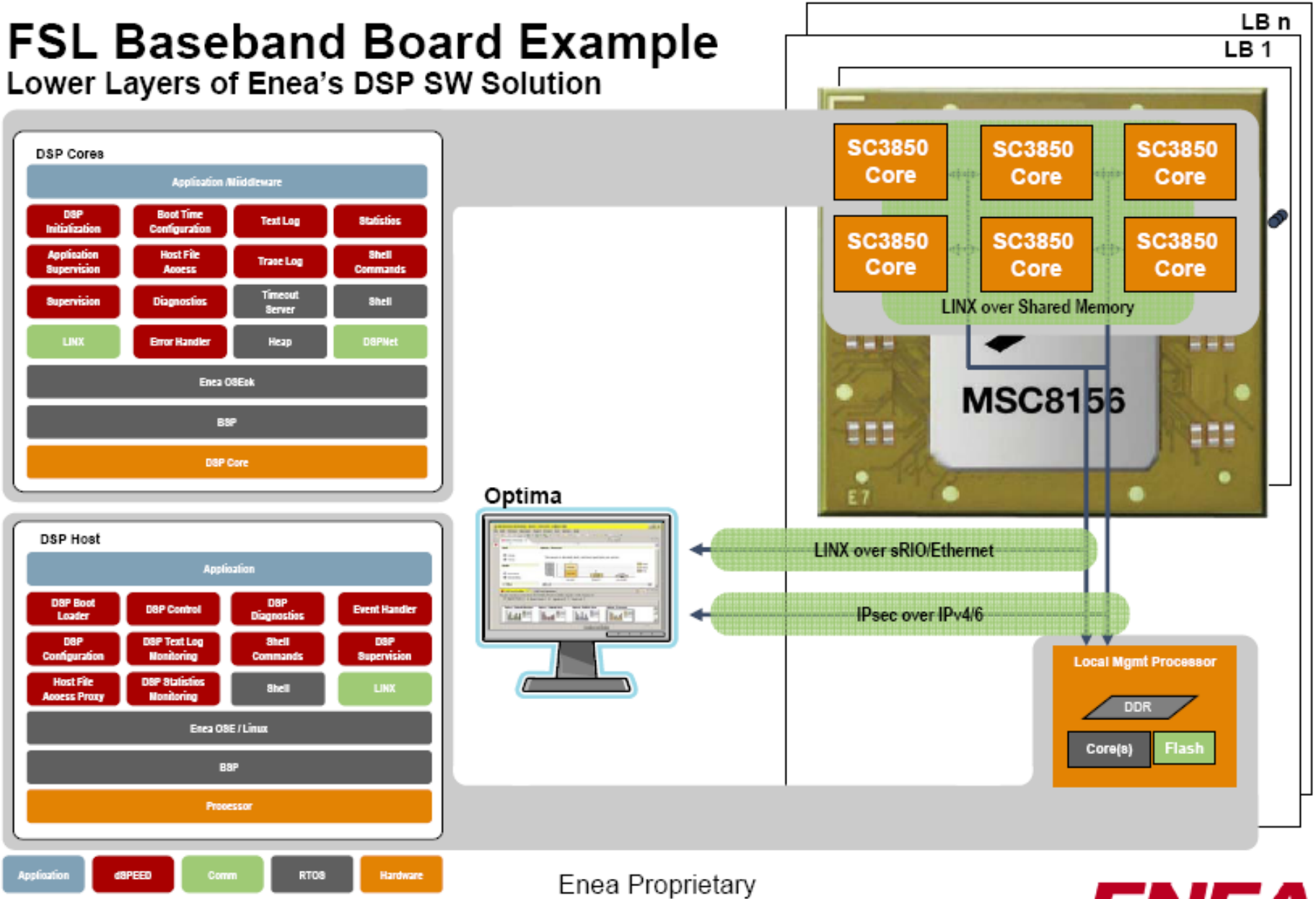


Enea Proprietary

ENEA

FSL Baseband Board Example

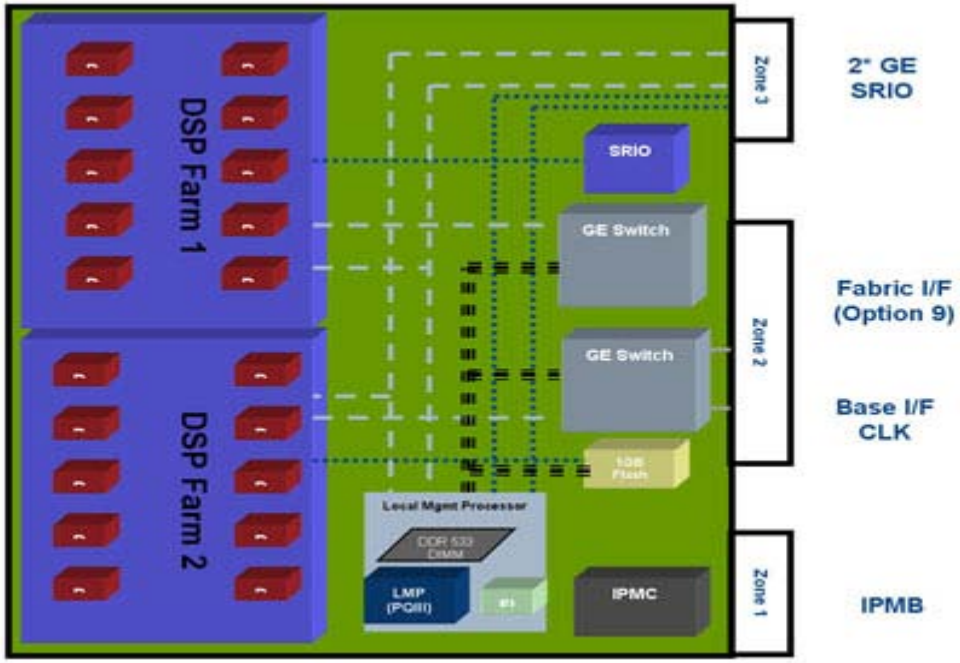
Lower Layers of Enea's DSP SW Solution



Enea Proprietary

ENEA

dSPEED Demo 1



- Each chassis is 8 DSP Boards
- Each board has 2 AMC Cards (Farms)
- Each Farm is 10 DSP
- Total is around: 8*2*10= 160 DSPs, and the Host is PPC based CPU and all of them running dSPEED.

dSPEED Demo 2

- **DSP AMC vard**
  - 4 x TI TCI6482 DSPs (C64x+ core)
- **CPU AMC card**
  - Freescale PQIII MPC8548 (e500 core)
- **AMC Carrier**
  - sRIO switch
  - Ethernet switch