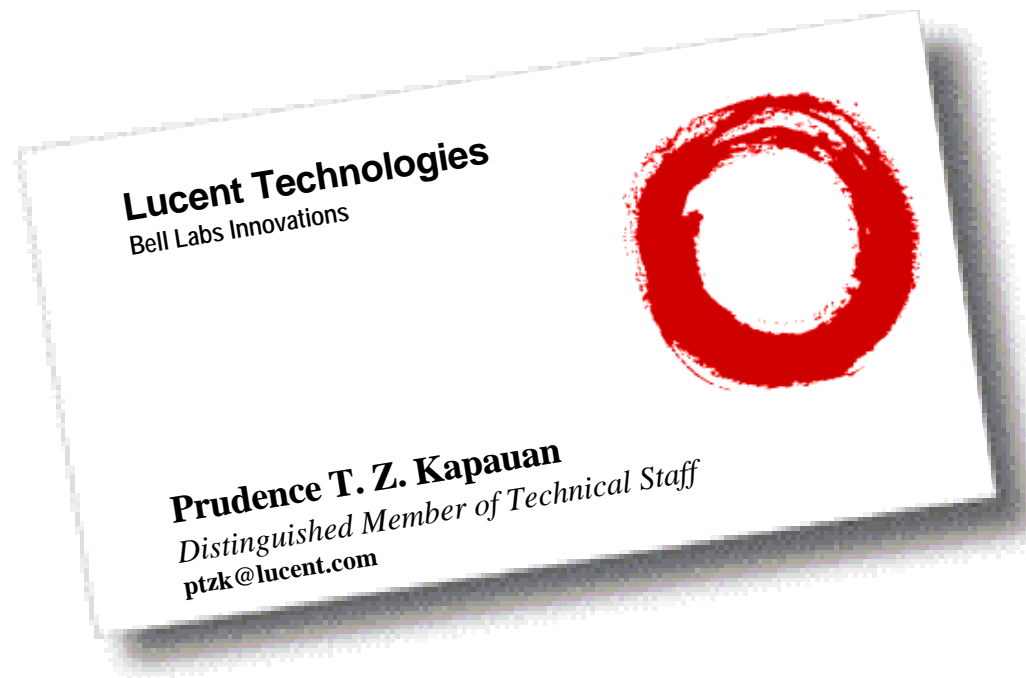


---

# Using Object-Oriented Technologies in SPEED



# Talk Overview

---

- What is SPEED?
- Why C++?
- How about CORBA?



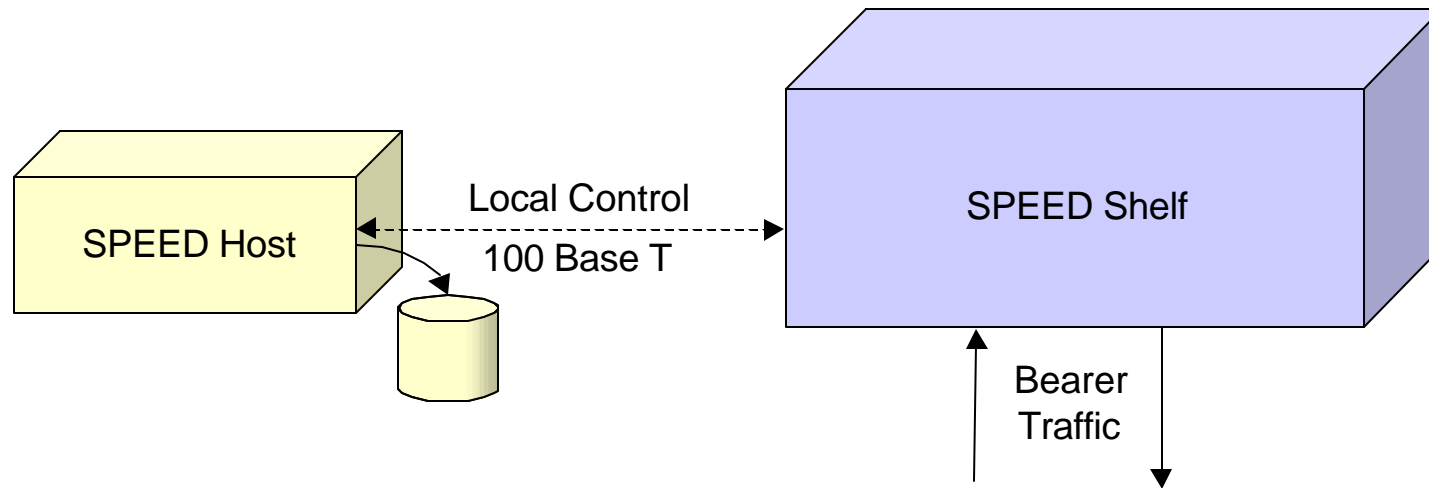
## What is SPEED?

---

- broadband, scalable switching product built on an open architecture and standard hardware and software interfaces.
- basis for many wireless applications in Lucent.
- suited for applications involving packet or circuit switching and/or processing of broadband, multimedia traffic.



# SPEED II Architecture – Configuration



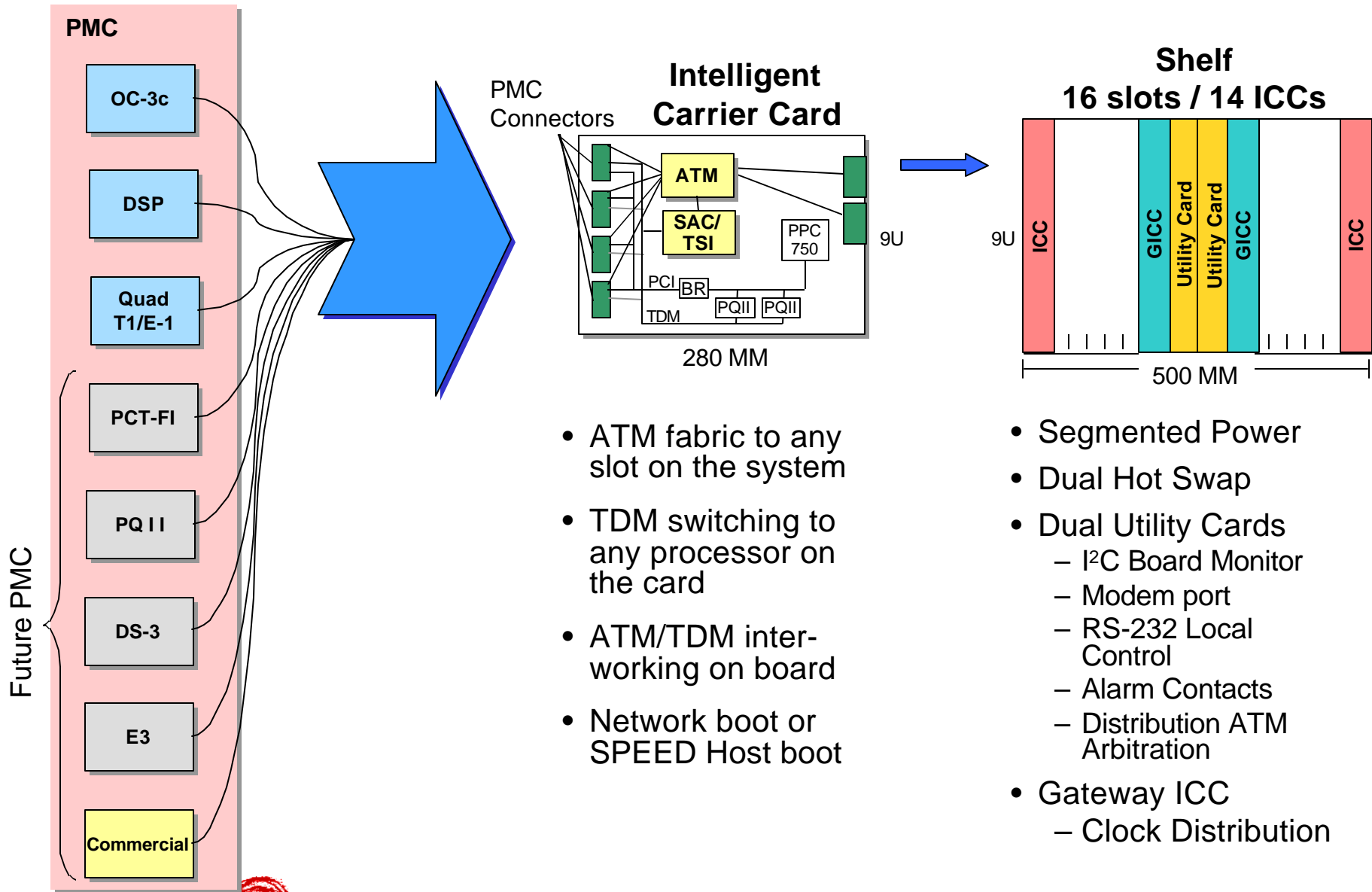
## SPEED Host

- Performs administrative tasks
- Provides User Interfaces
- Stores Provisioning data and master software images

## SPEED Shelf

- Fully redundant (highly reliable)
- Processes bearer channel traffic
- Continues operating if SPEED Host is temporarily unavailable

# SPEED II Hardware – Shelf Components

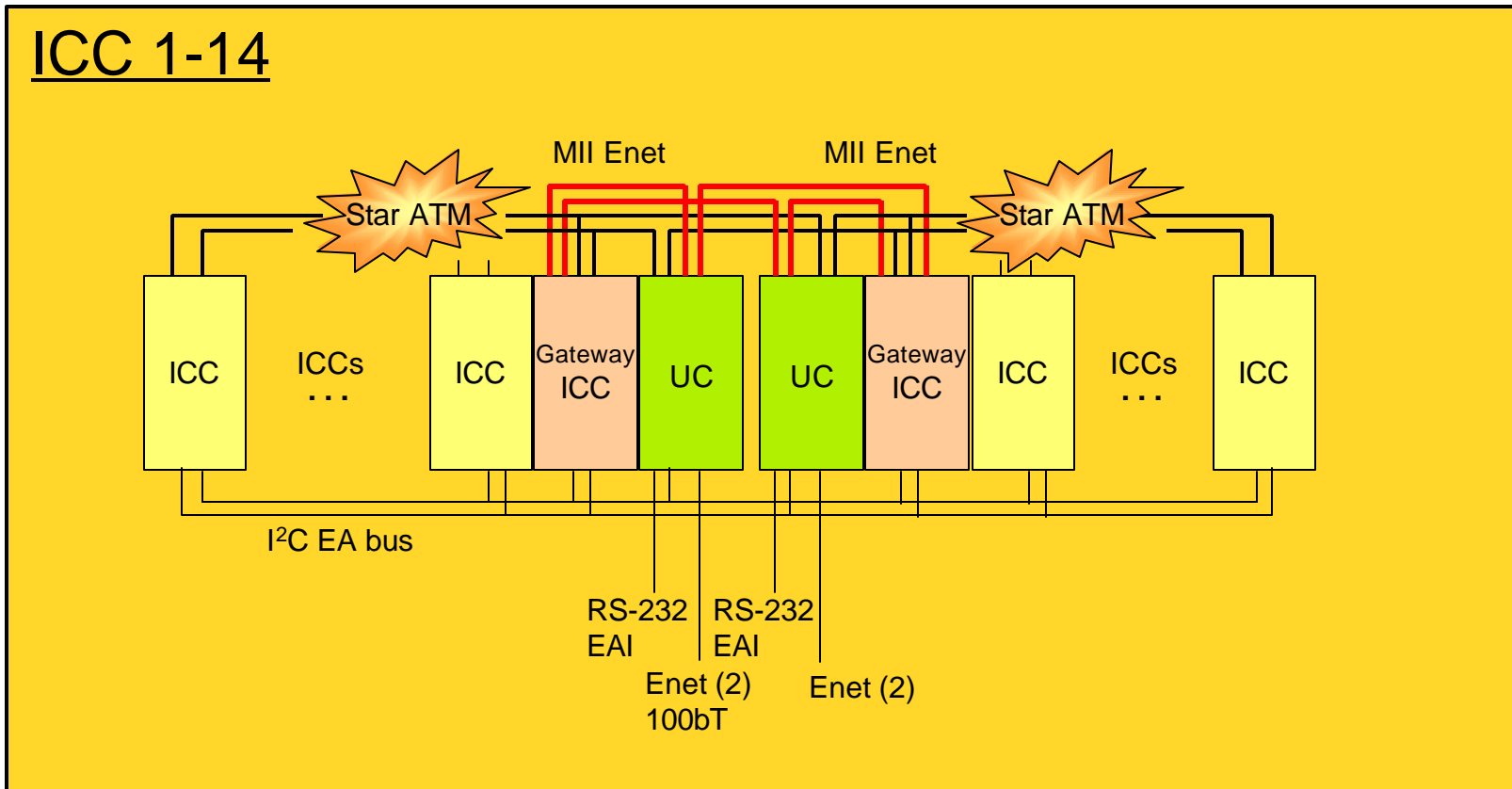


- ATM fabric to any slot on the system
- TDM switching to any processor on the card
- ATM/TDM inter-working on board
- Network boot or SPEED Host boot

- Segmented Power
- Dual Hot Swap
- Dual Utility Cards
  - I<sup>2</sup>C Board Monitor
  - Modem port
  - RS-232 Local Control
  - Alarm Contacts
  - Distribution ATM Arbitration
- Gateway ICC
  - Clock Distribution



# SPEED II Hardware – Intelligent Carrier Card (ICC)



- GICC = Gateway ICC
- UC = Utility Card
- MII = Media Independent Interface

## SPEED II Hardware – PCI Mezzanine Cards (PMCs)

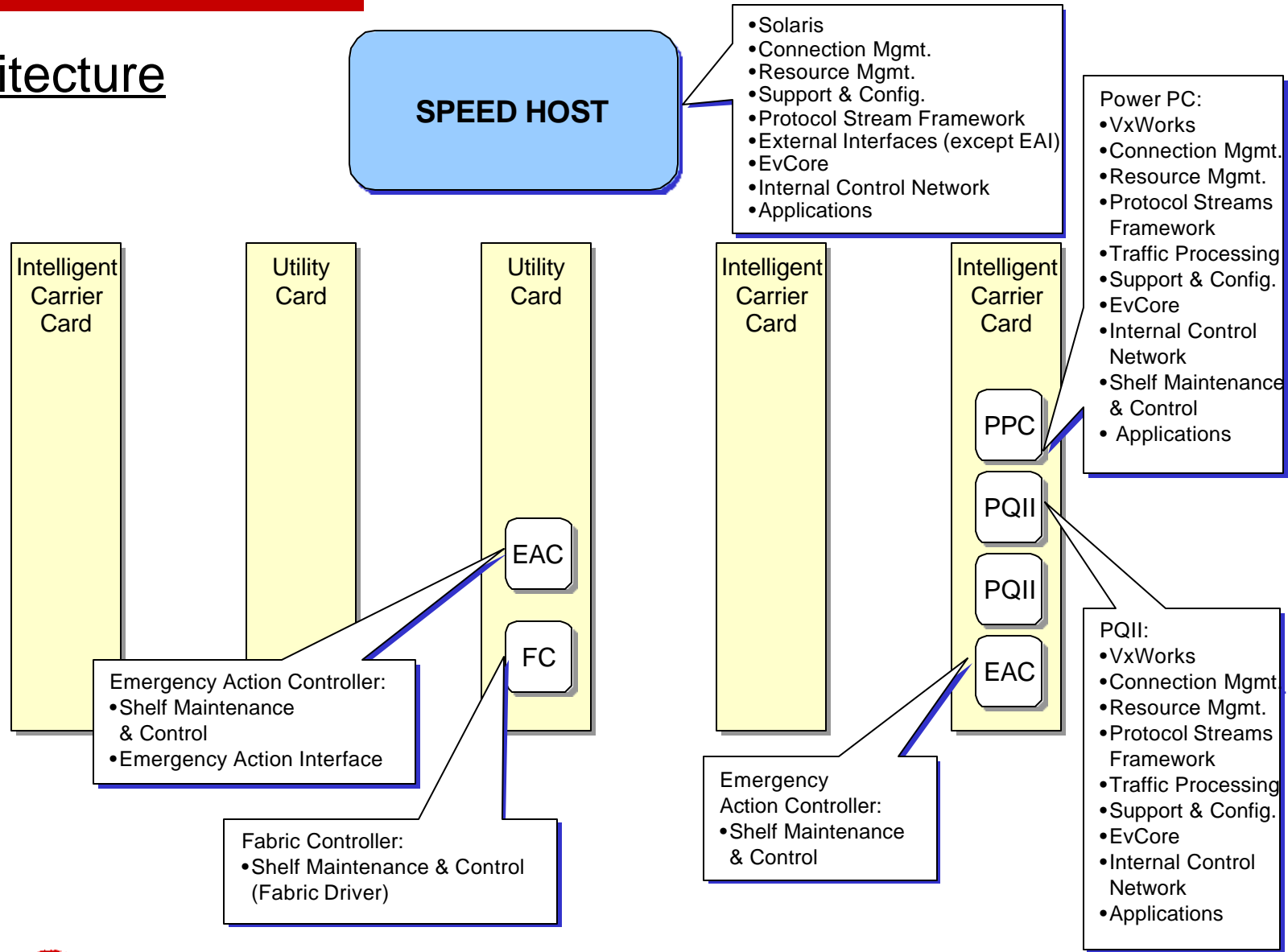
---

- 4 PMC slots in ICC
- Physical interfaces (e.g., ATM, DS-1/E-1)
- Processing Power (e.g., DSP)



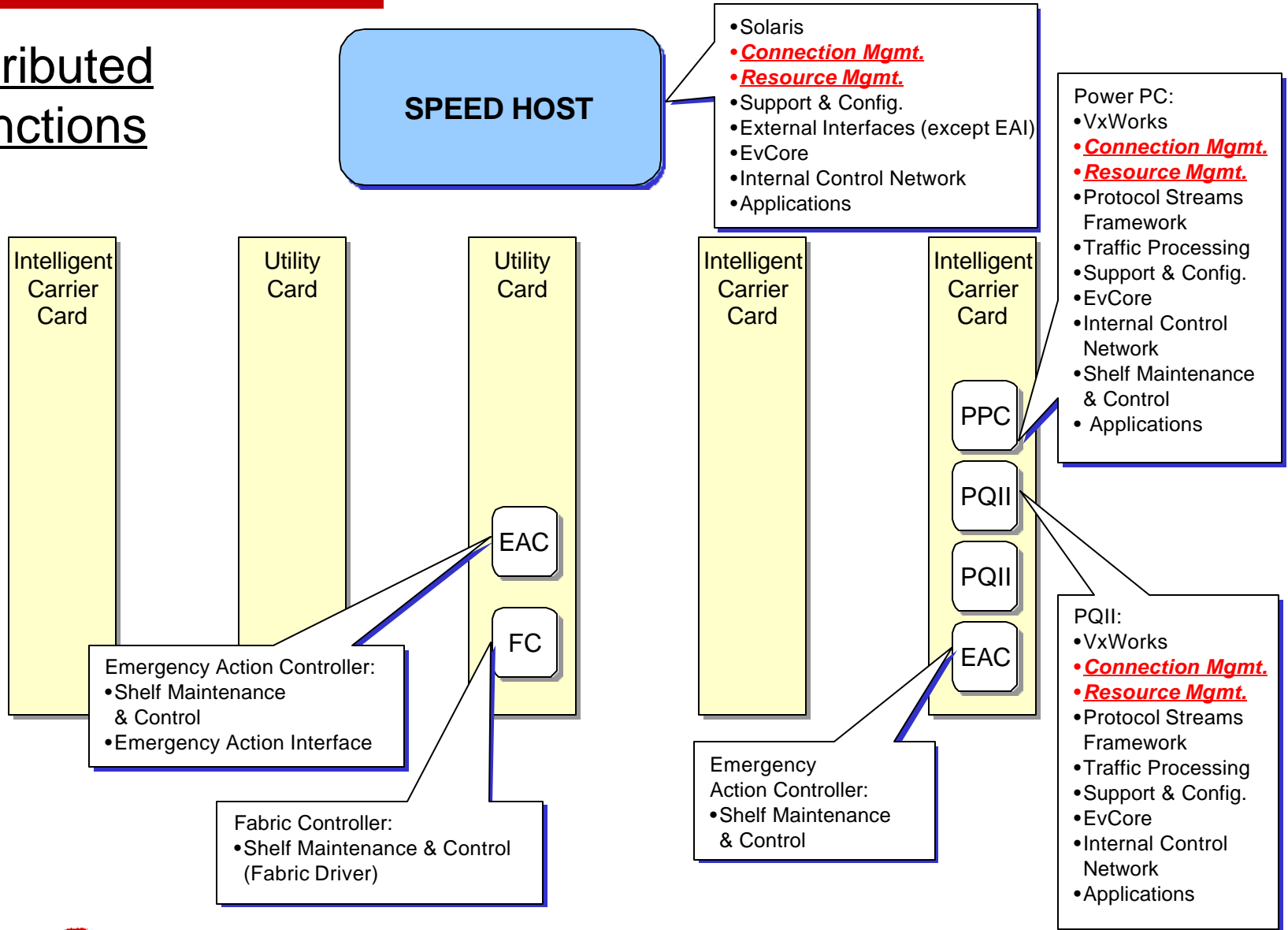
# SPEED II Software – Platform Software Mapping

## Architecture



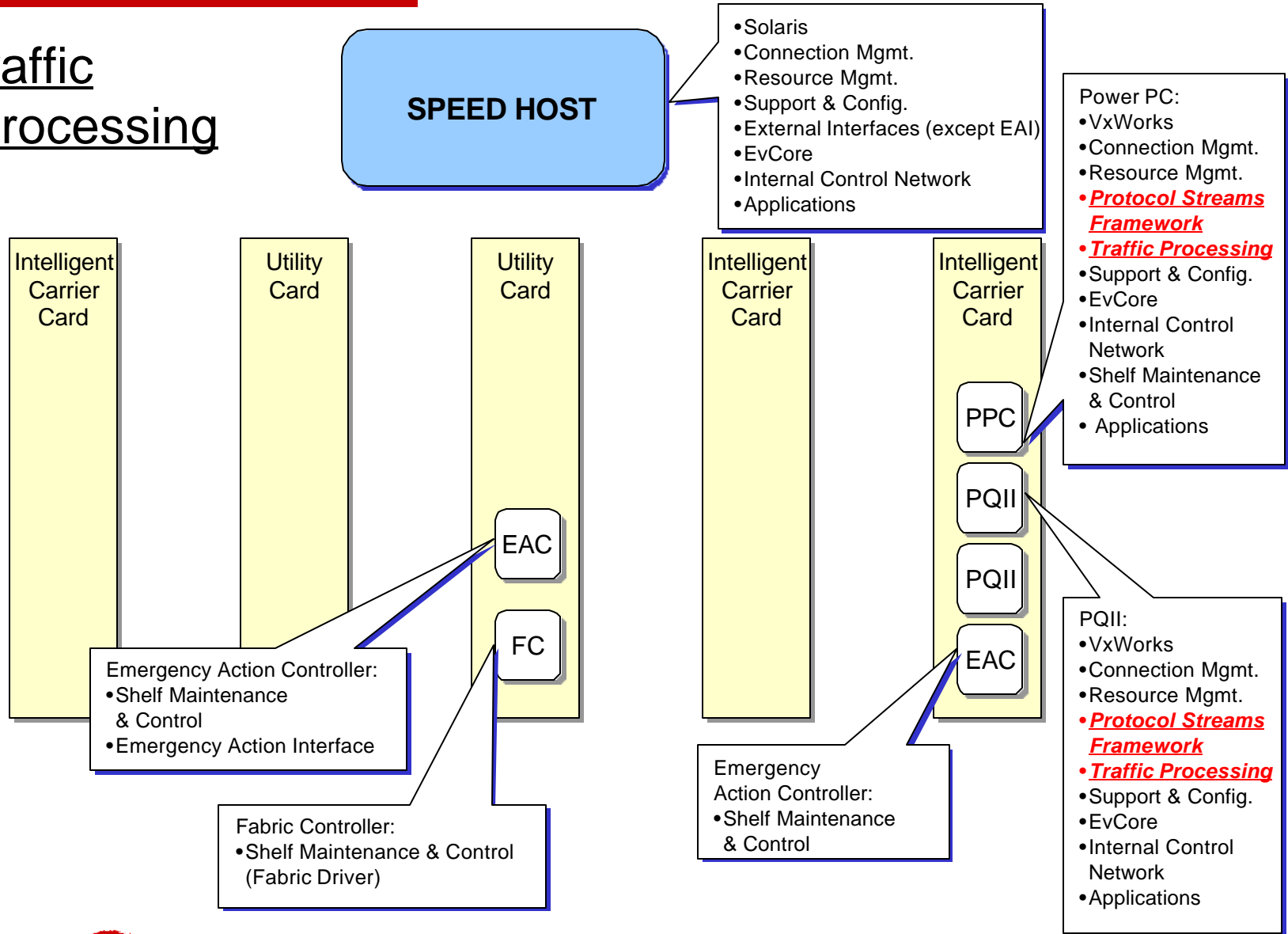
# SPEED II Software – Platform Software Mapping

## Distributed Functions



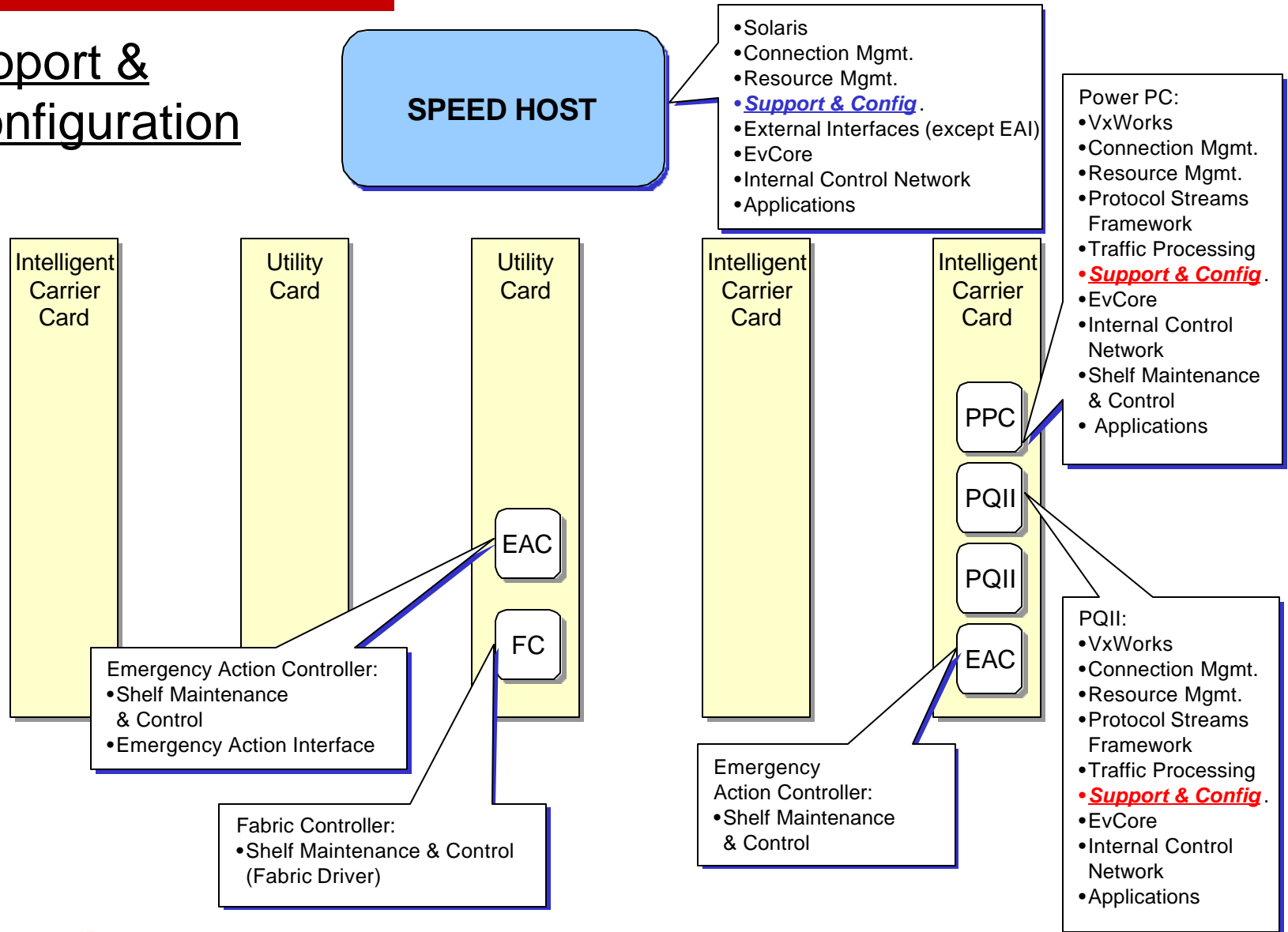
# SPEED II Software – Platform Software Mapping

## Traffic Processing



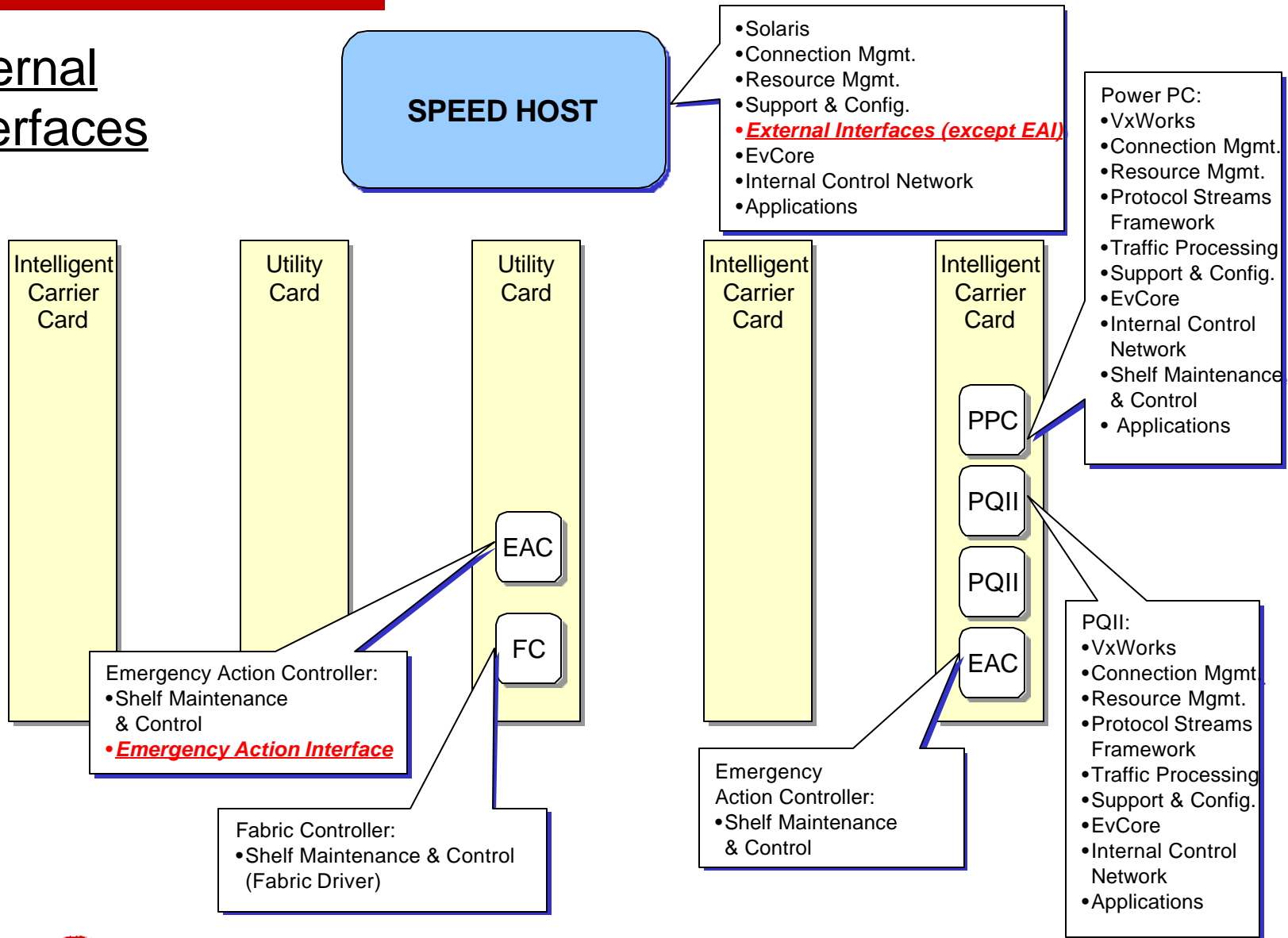
# SPEED II Software – Platform Software Mapping

## Support & Configuration



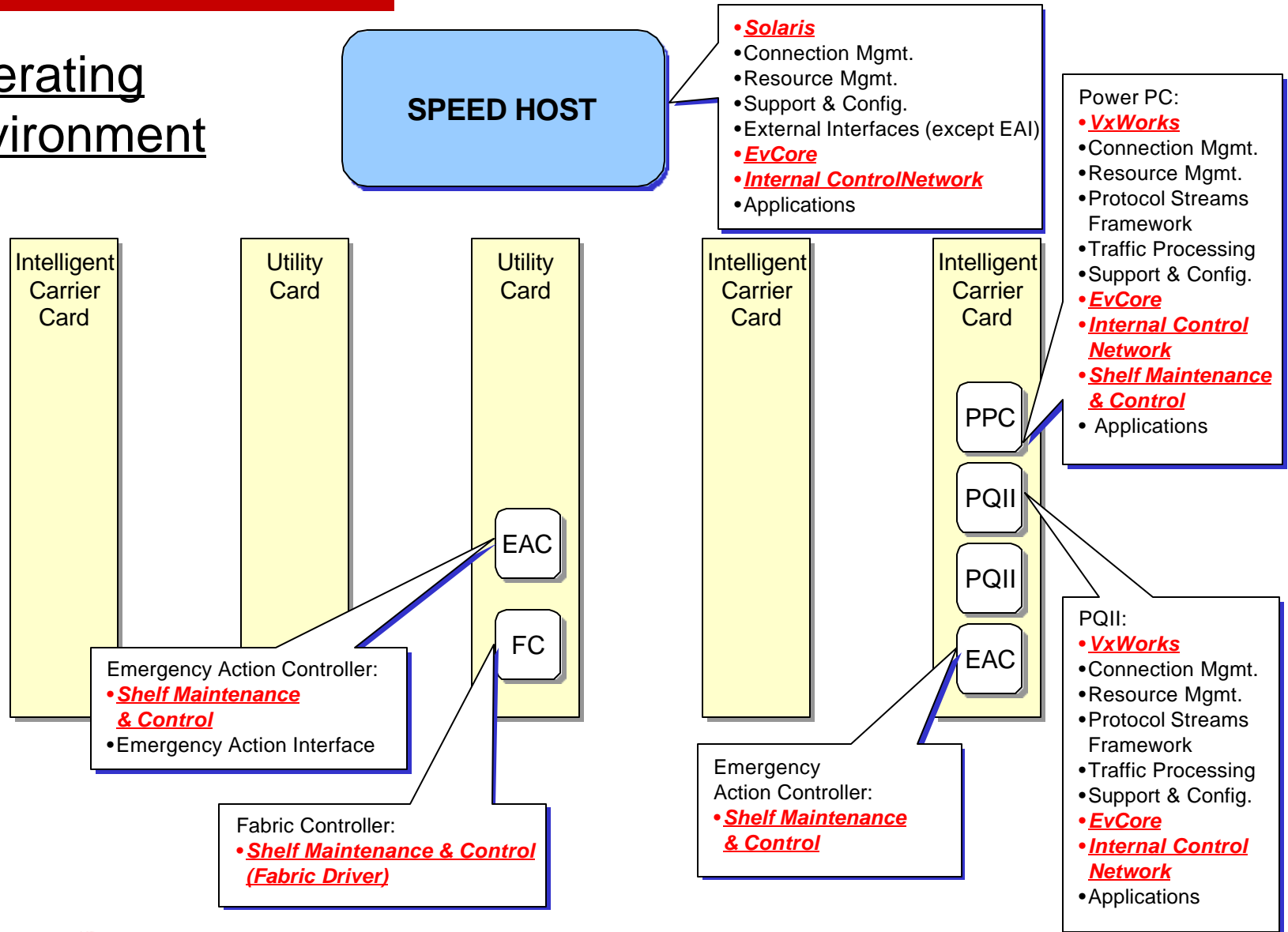
# SPEED II Software – Platform Software Mapping

## External Interfaces



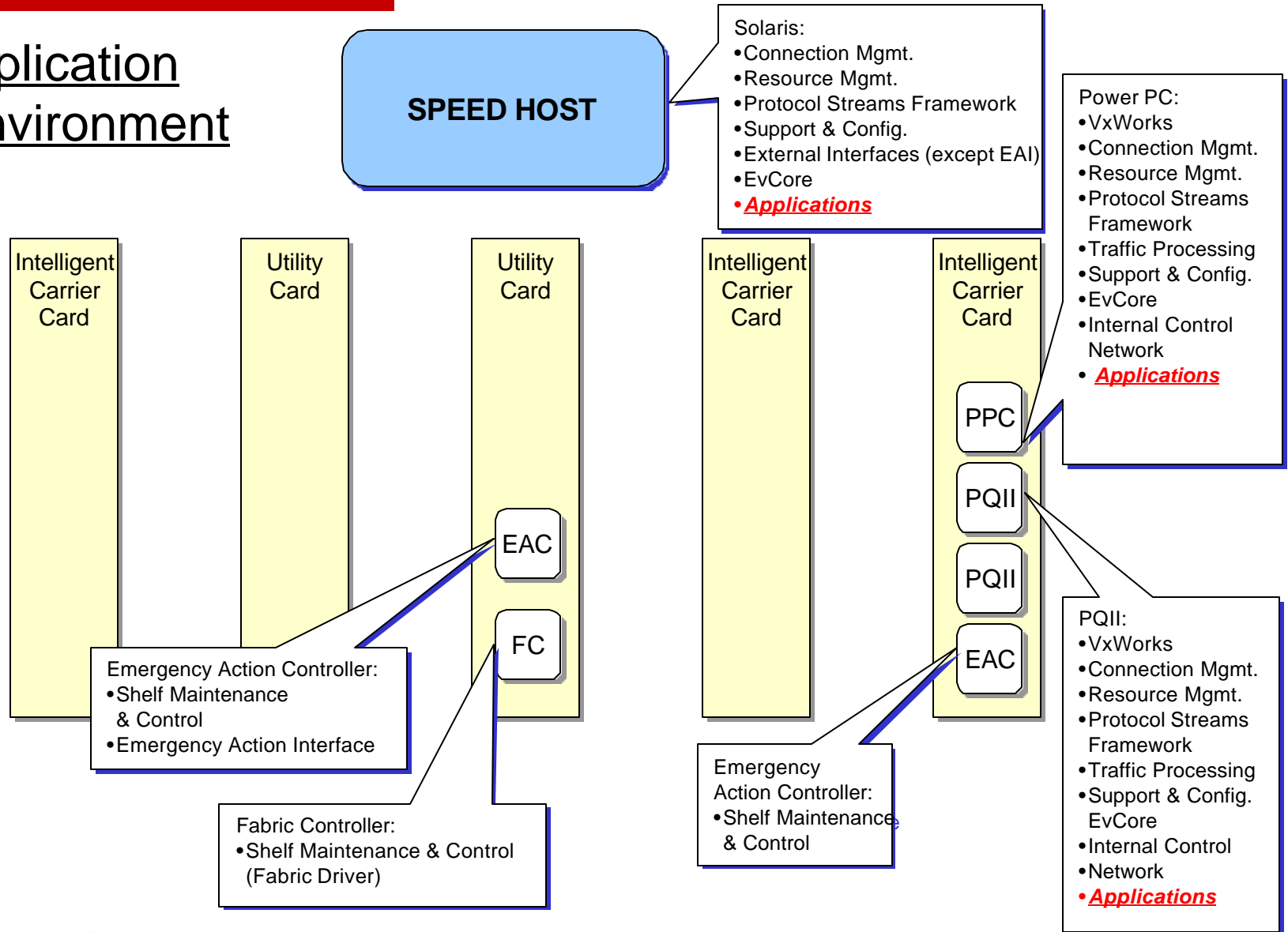
# SPEED II Software – Platform Software Mapping

## Operating Environment



# SPEED II Software – Platform Software Mapping

## Application Environment



# Why C++?

---

Ref: EE Times Special Issue on Embedded Systems 9/25/00

- 3 major factors for language selection:

- unification of design flow elements
- speed of use provided by the language
- evolution for older technologies

- In addition:

- natural mapping into object classes
- ease of collaboration among developers



# Connection Management Framework

---

- Provides applications with life cycle control of connections
- Views system as a set of computing and I/O resources and a set of connection domains
- Object classes:
  - Connection Manager (singleton pattern, one per processor)
  - Switch Fabric Manager (one per switching domain)
  - Trunk Manager (connects switching domains)
  - Service Access Point (connection endpoint)



# Resource Management Framework

---

- Responsible for resource naming and states:
  - Resource naming and inventory
  - State interrogation and reporting
  - State change requests
  - Driver control to map states to resources
  
- Object classes:
  - Maintainable resource
  - Message class

# Data Management Framework

---

- Serves processes with configuration and/or persistent data.
- Consists of a DMF server and a Client Library
- Object classes:
  - Container class
  - Transaction objects
  - Registration objects



## EvCore Framework

---

- Provides facilities for creating event-driven applications or other frameworks.
- Loosely coupled collection of code modules composed of one or a few related C++ classes.
- Code Modules and Object classes:
  - Event and Event Queue classes
  - Timer and Timer Manager classes
  - I/O Select and Select Manager classes
  - Kernel Modules and Classes
  - Signal Handling Classes



## How about CORBA?

---

- Prototyping work
- Future Work



## Prototyping Work: Protocol Streams Framework

- Streams bearer traffic through a collection of channel processing modules
- Original version used a subset of Adaptive Communication Environment (ACE) Framework
  - Adaptive Service Executive (ASX) Stream Framework
- Prototype implementation to do wireless protocol processing
  - Very demanding in performance, efficiency
- Stream Framework (based on Unix System V Release 4 streams)



## Future Work: Next Generation Platform

---

- Scalable capacity to 2.5Tb/s
- Scalable availability to 5 9's or better
- Multi-protocol support for interfaces
- Highly modular and configurable
- Advanced software infrastructure
- Explore the use of Fault Tolerant CORBA to provide the functions of the Fault Detection and Recovery Framework

