

Carrier Grade Linux: A to Z!

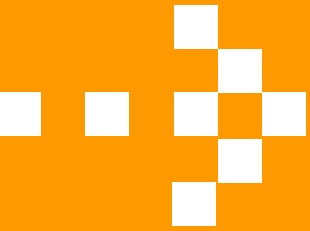
The full story, or *almost* ...

Ibrahim Haddad, OSDL

Ottawa Linux Symposium
July 2005

Agenda

- Linux – Disruptive Technology
- Telecommunications Market
- Carrier Grade Linux
- Deployment
- Conclusion



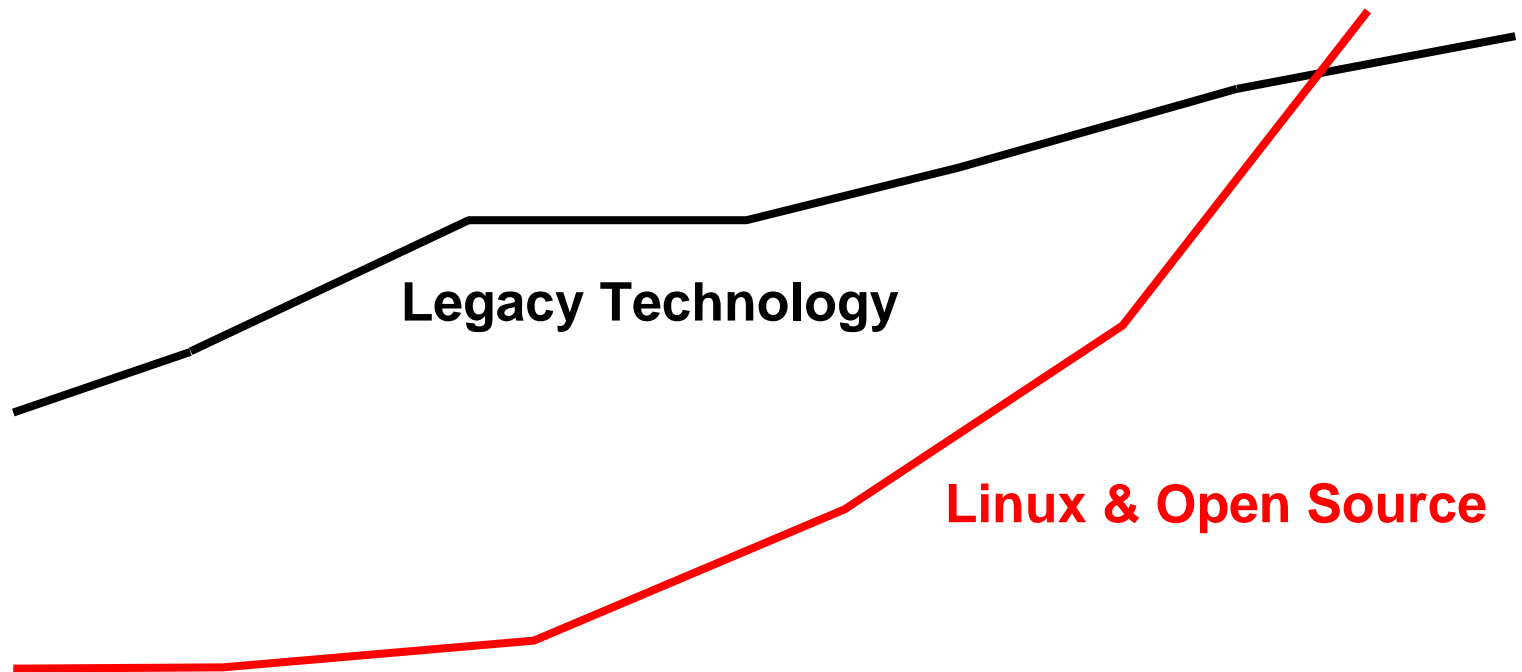
Why companies adopt Linux?

Why Adopt Linux and Open Source?

- GPL – Full access to source code
- Platform independence
 - Lower hardware costs and extended system life-cycle
- Lower Maintenance and Support
 - Minimal system administration support required after installation
- Pricing
 - Lower costs vs. legacy / proprietary OS
 - Entire supply chain looking to lower TCO
- Multiple providers
- Higher system performance, reliability and security
- Source code quality, Innovation rate, peer reviews and testing resources
- Established eco-system: Hardware and Software

Linux is a Disruptive Technology. Really.

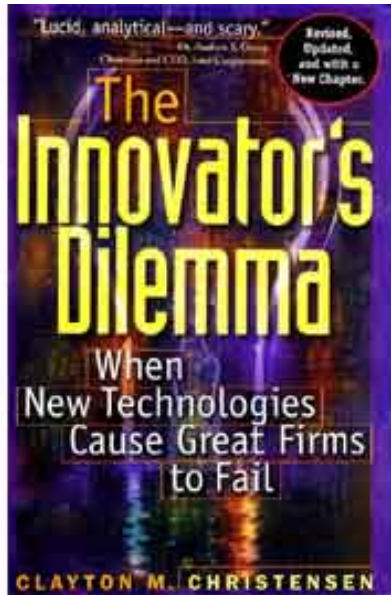
Linux can't do [fill in blank] is an invitation to Open Source developers



Disruptive Technologies

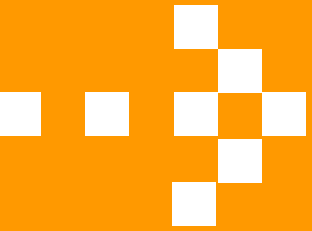
- What is disruptive technology ?
 - Technology with significant deficiencies that is targeted to a niche segment but providing significant cost benefits
 - Usually targeted to early adopters that are the ‘risk takers’ with a much slower adoption into the mainstream
- Disruptive technologies in telecommunication
 - Voice Over IP
 - Linux Operating System
- Adoption of Disruptive technology always starts with non mission-critical applications

Disruptive Technologies



The Innovator's Dilemma
Clayton M. Christensen

- Reinvent Business Models
- Decentralize Vertical Markets
- Clarify Market Vision
 - *Linux/Open Source help identify real value-added*
 - *Those who do not provide clear value are automatically out of the market*



The Telecom Market

Yesterday and Today

Yesterday ...

Communications and data service networks built on proprietary platforms to meet specific requirements for

- Availability
- Reliability
- Performance
- Service response time

Today ...

Communications service providers must drive down cost while maintaining carrier-class platforms with:

- High availability
- Scalability
- Security
- Reliability
- Predictable performance
- Maintenance & upgrade

Why the attention to Linux from Telecom?



Service Provider

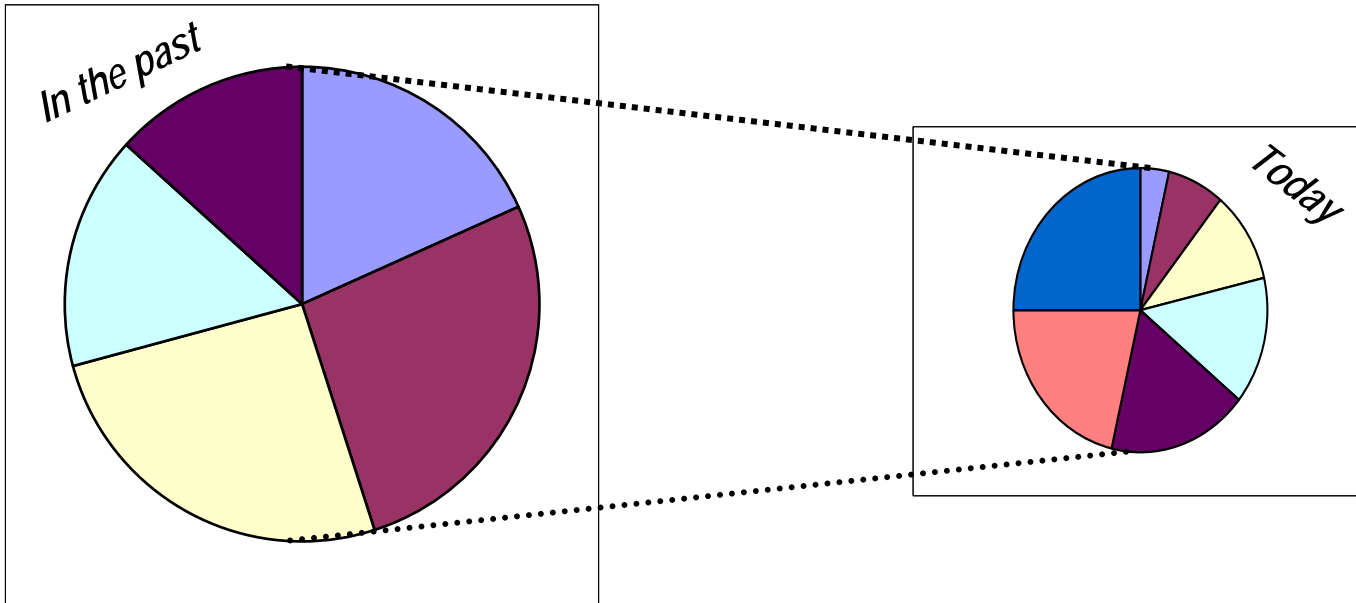


Proprietary/Closed Technology



Copyright © 2003 United Feature Syndicate, Inc.

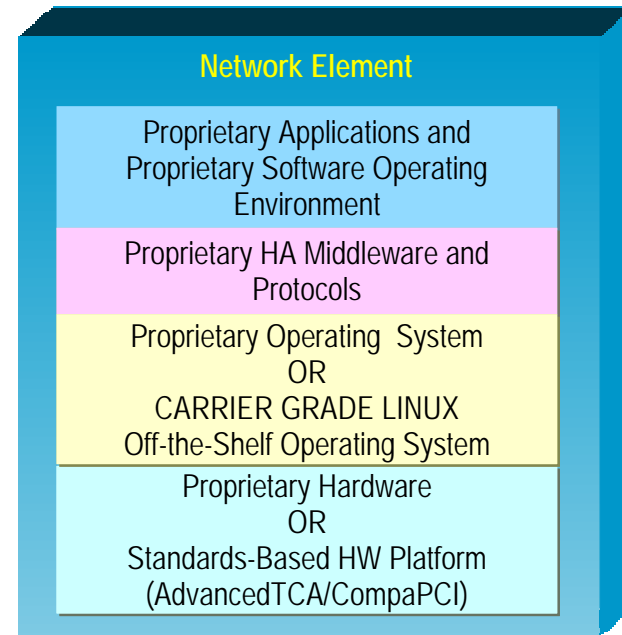
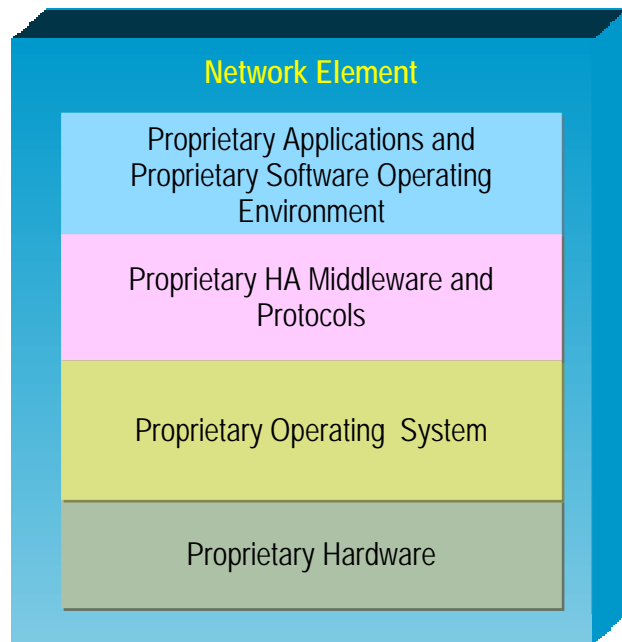
Telephony Business in (R)evolution



- Voice telephony is a premium service.
- Few players
- Profits

- Voice telephony is a commodity
- More players in the field
- Shrinking Business – fierce competition and pricing models
- Much less profits
- Losses in some areas
- More threats (VoIP, Broadband telephony)

Telecom Platform (R)evolution



PAST

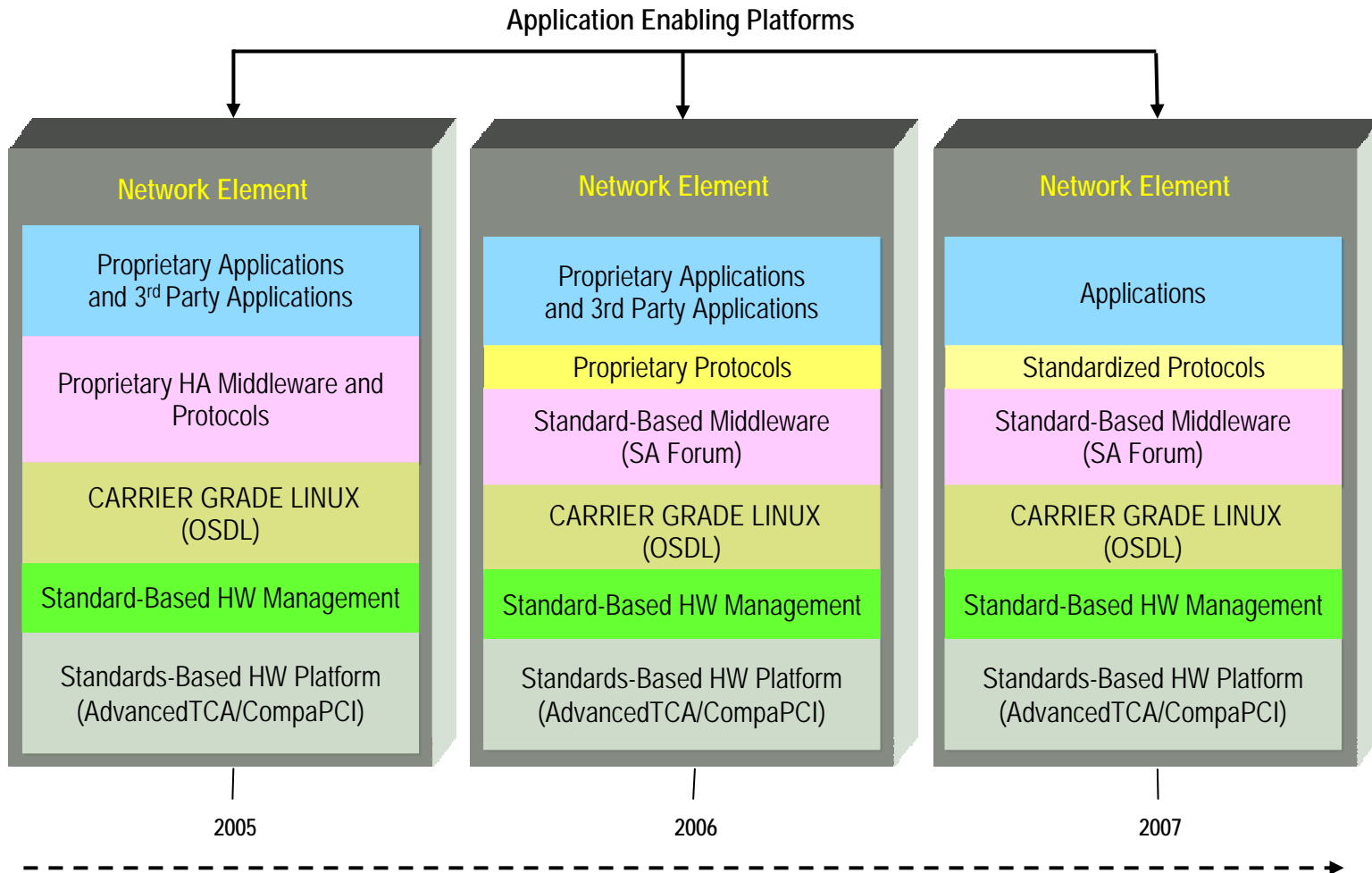
- Proprietary/Legacy systems
- No clear separation of “building blocks”
- Proprietary technologies & interfaces
- Expensive to develop, maintain & scale
- Single Provider

- Open standard-based systems
- Interchangeable “building blocks”
- Standardized interfaces
- COTS SW and HW
- Multiple Providers

Why this shift, now?

- Service providers and carriers are in a position today where they must move away from specialized proprietary architectures, and towards COTS approaches and building practices driven by two key motivations:
 - **Faster time to market:** They need to be able to deliver new services based on common standardized platforms. They are in a constant race to deliver faster to the market.
 - Building with proprietary and specialized technologies that are offered by a very limited number of providers is one obstacle from this perspective.
 - **Reduce costs:** They need to reduce the design and operation costs by using COTS hardware and software components
 - COTS components are offered by multiple providers
 - COTS components are compliant or registered towards standards of industry agreed specifications.

Telecom Platform (R)evolution

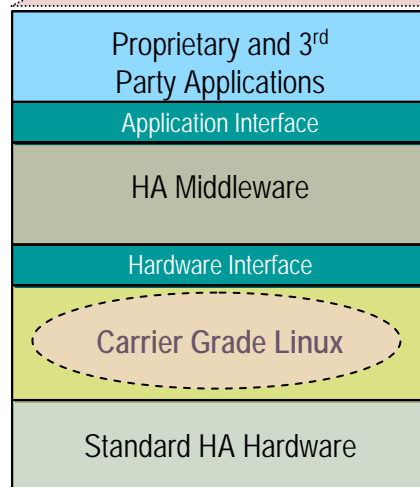
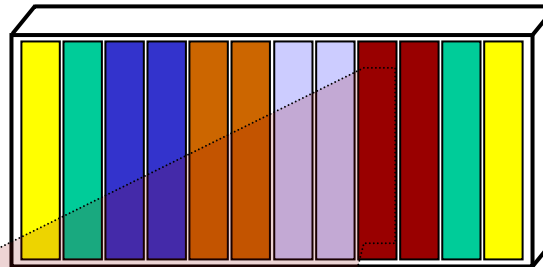


Rapid Adoption Promotes Industry Innovation – Resources are focused on new applications

The Results ...

Network Element
Interchangeable standard
based building blocks with
standardized interfaces

One subrack
Multiple Network Elements



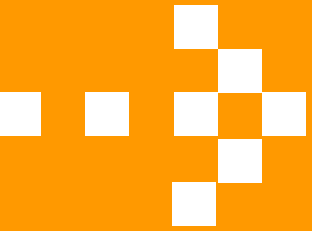
Challenges

- **Reduce Costs**
 - Using COTS building blocks (SW and HW)
- **Seamless integration** of carrier-grade components
 - Integrated solution must be validated for carrier-grade availability
- **Maintain carrier-class characteristics**
 - Delivering **increasing levels of availability** and dependability
 - **Growth of packet traffic** putting pressure on communication networks
 - Platforms in all-IP environment must maintain their carrier-class characteristics
- **Decrease time to market**
- **Fast delivery of new services** by shorten new service dev time
 - Unifying platforms
- **Increase profits!**

Don't Shoot the Messenger!



- Not the doom of NEPS and Carriers
 - Not “turning the industry on its head” either
- Instead a “sea change”
 - Transition to COTS architectures and practices
 - **Embrace of Linux and Open Source**
 - Re-alignment at multiple levels
 - **Before 1999/2000:** incompatible platforms, protocols, etc., high barriers to entry, circuit switched,
 - **Today:** Telecom resurgence with COTS, Linux & OSS
 - New players, new businesses
- **Key to success is understanding difference between core value add and marginal business**



Carrier Grade Linux

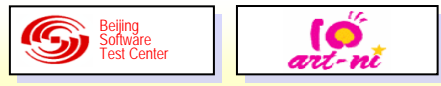
CGL WG

An industry forum to support & accelerate the development of Linux functionality for telecommunication applications

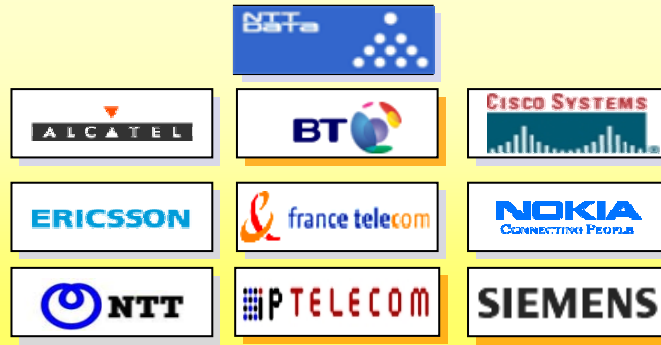
End-Users



Integration & Services



TEMs, NEPs, Carriers & Device OEMs



Applications



Middleware



System S/W & OS Distros



Systems



Silicon

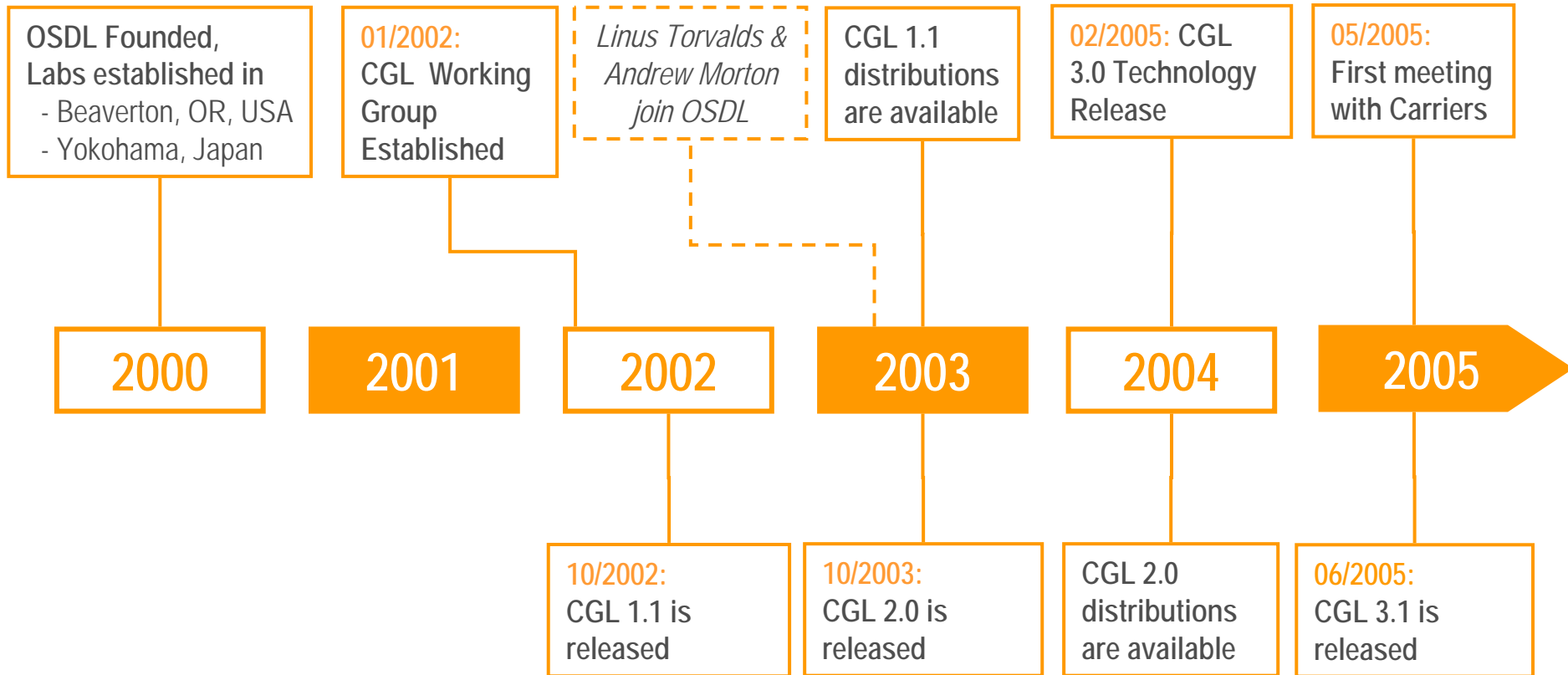


CGL Goal

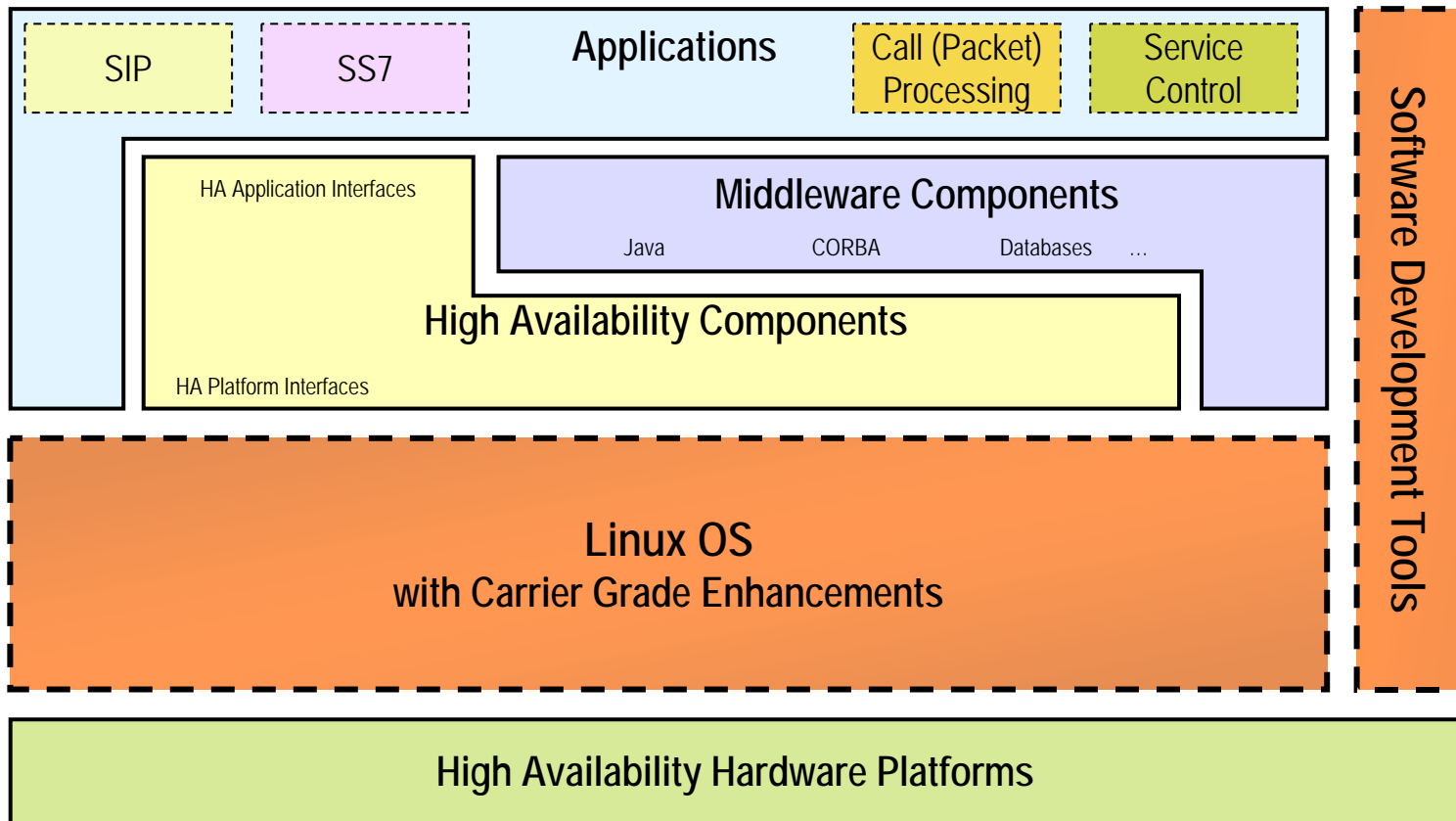
- **Goal:** Making Linux Better for the Telecom Industry
- **Vision:** Next generation and multimedia communication services are delivered using Linux-based open standard Carrier Grade Platforms.

A Linux kernel with Carrier Grade characteristics is an essential building block component for telecom platforms and architectures.

CGL History

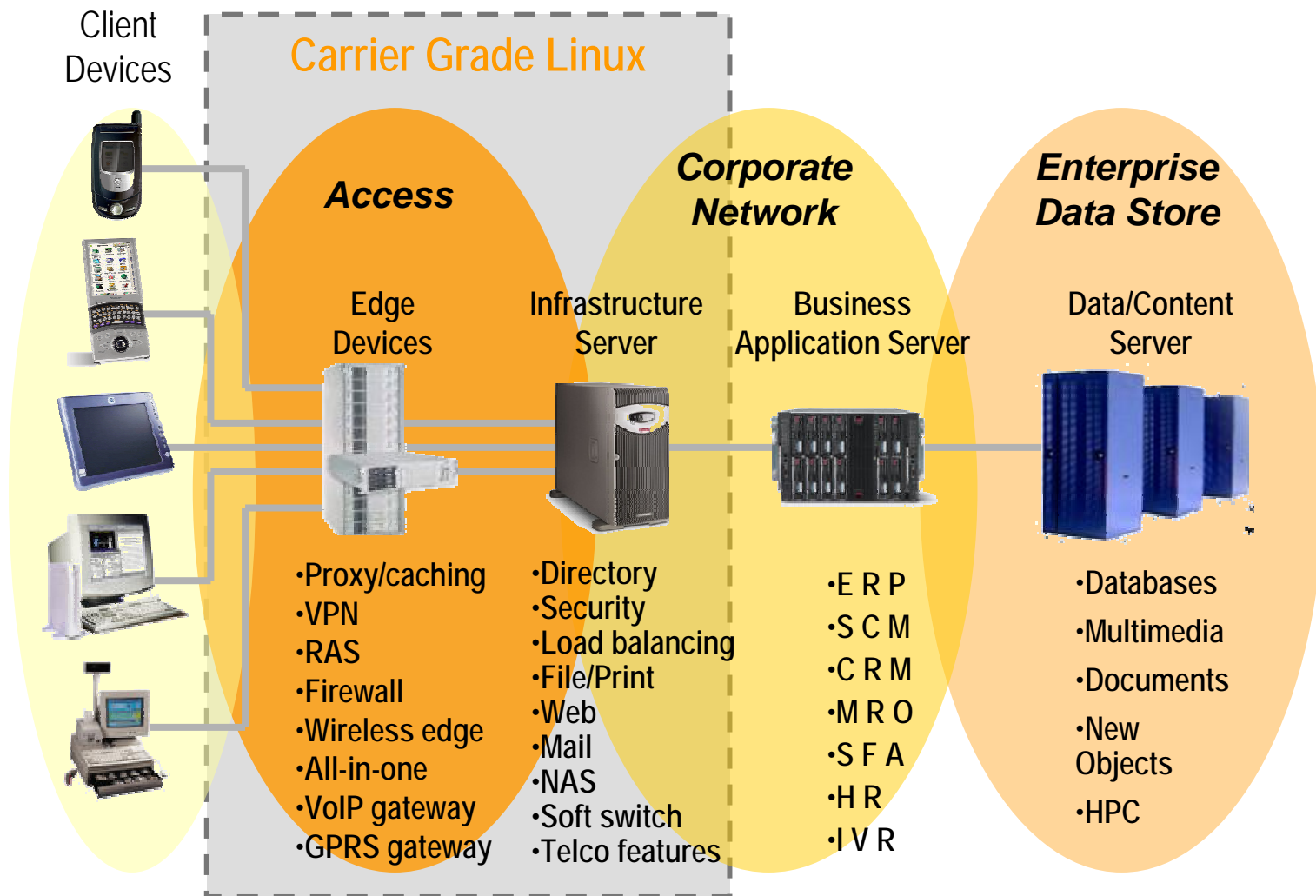


Carrier Grade Linux – Scope

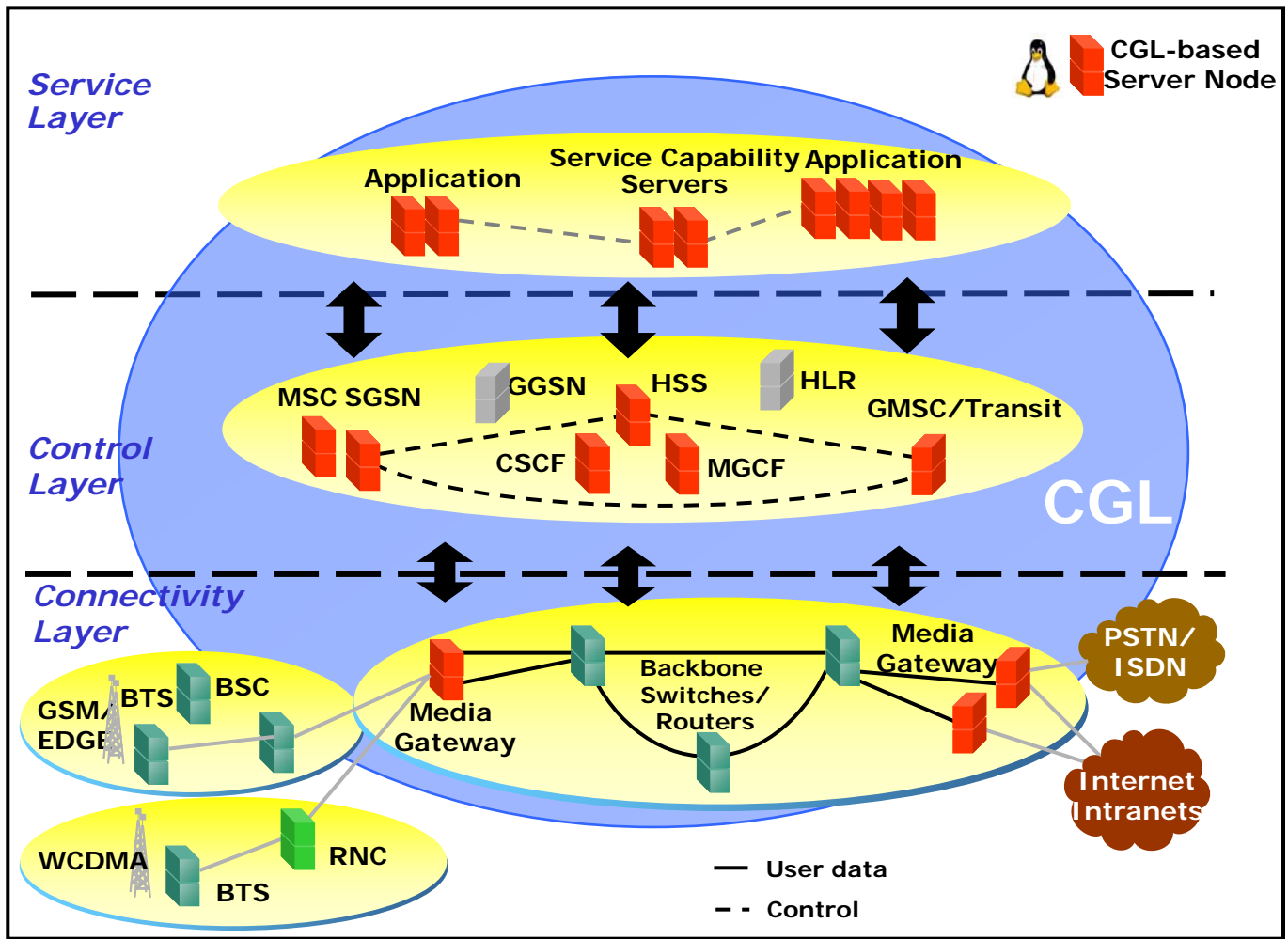


 *Scope of the Carrier Grade Linux Working Group*

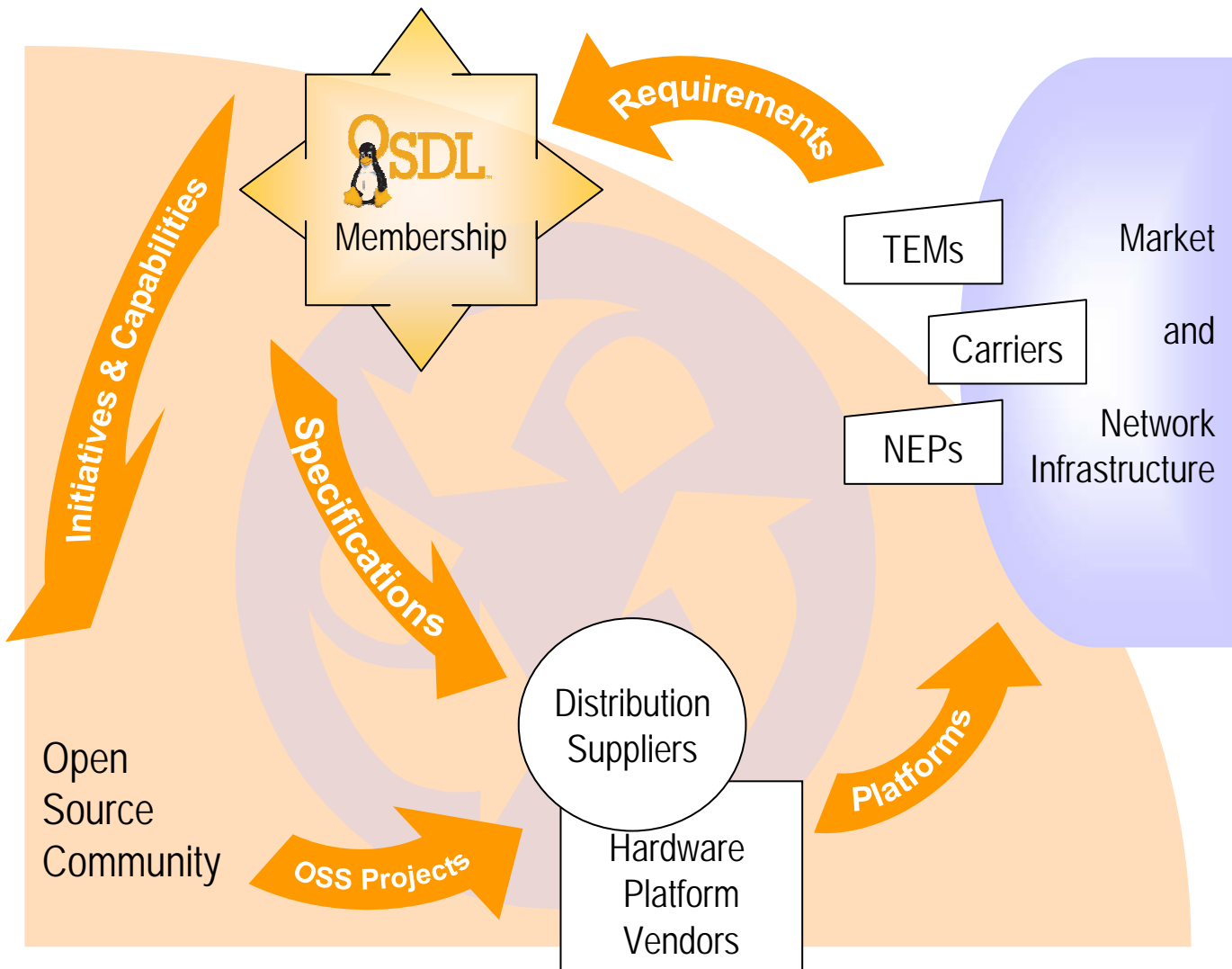
CGL Context – IT Networks



CGL Context – Next Generation Networks



Carrier Grade Linux – Lifecycle Process



CGL 3.1 Requirements Areas



- Standards-compliance / APIs



- Hardware Support



- Availability



- Clustering



- Security



- Performance



- Serviceability

Current Activities

- Focusing on development
 - Concentrate on promoting quality implementations of the 3.1 spec in 2005-2006
 - Resources on the CGL tech board are being realigned accordingly
 - New SIGs established
- Extend gap analysis and use case descriptions
- Refine CGL specification
- Document carrier inhibitors in MRD
- Work on specific focus areas (next slide)

Beyond CGL 3.1 - Specs

First half of 2005



Current Specification Structure
Separate Document per Focus

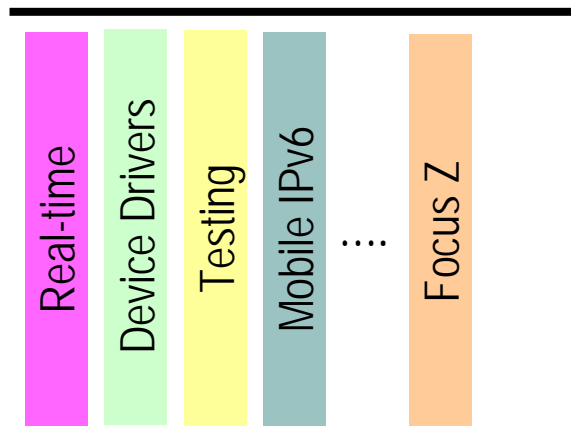
New Focus Areas



Updates to previous Focus Areas



July 2005 onward



- Smaller, more focused releases.
- Break out deep technical items.
- Separate new work from updates.

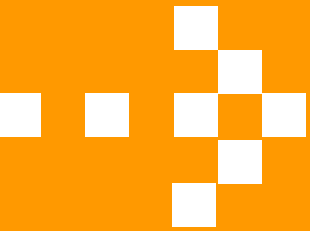


Special Interest Group – Focusing on Dev Efforts

- Existing SIGs
 - Storage SIG
 - *Participants:* OSDL, IBM, Novell, EMC, RH, Unisys, Bull, SUN, DreamWorks, HP, Cicso, Intel
 - Hotplug:
 - *Participants:* OSDL, HP, Virtual Iron, Fujitsu WindRiver, Intel , Bull, Unisys, EMC, SGI
 - Security
 - *Participants:* OSDL, Novell, IBM, Sun, Ericsson, MontaVista, HP, SE Linux, NCSC
 - Clusters
 - *Participants:* OSDL, RH, IBM, HP, Novell, MontaVista, SUN, Intel, Oracle, ORNL
- New SIGs:
 - Robust mutexes
 - Live patching S
 - System management

End User Forums

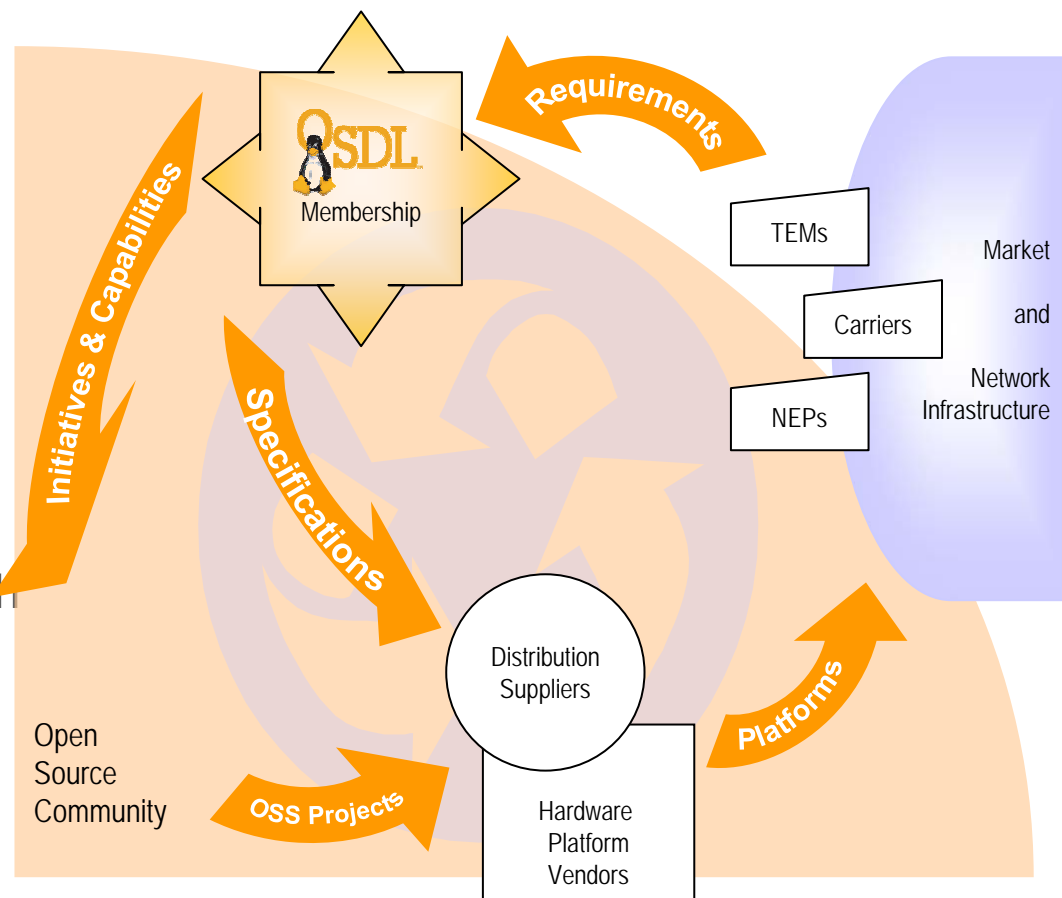
- Carriers Forum
- Linux User Advisory Councils (LUACs)
- Independent software vendors (ISV forum)
- Independent hardware vendors (Open Source Driver Forum)



CGI and Src Code

Sequence of events

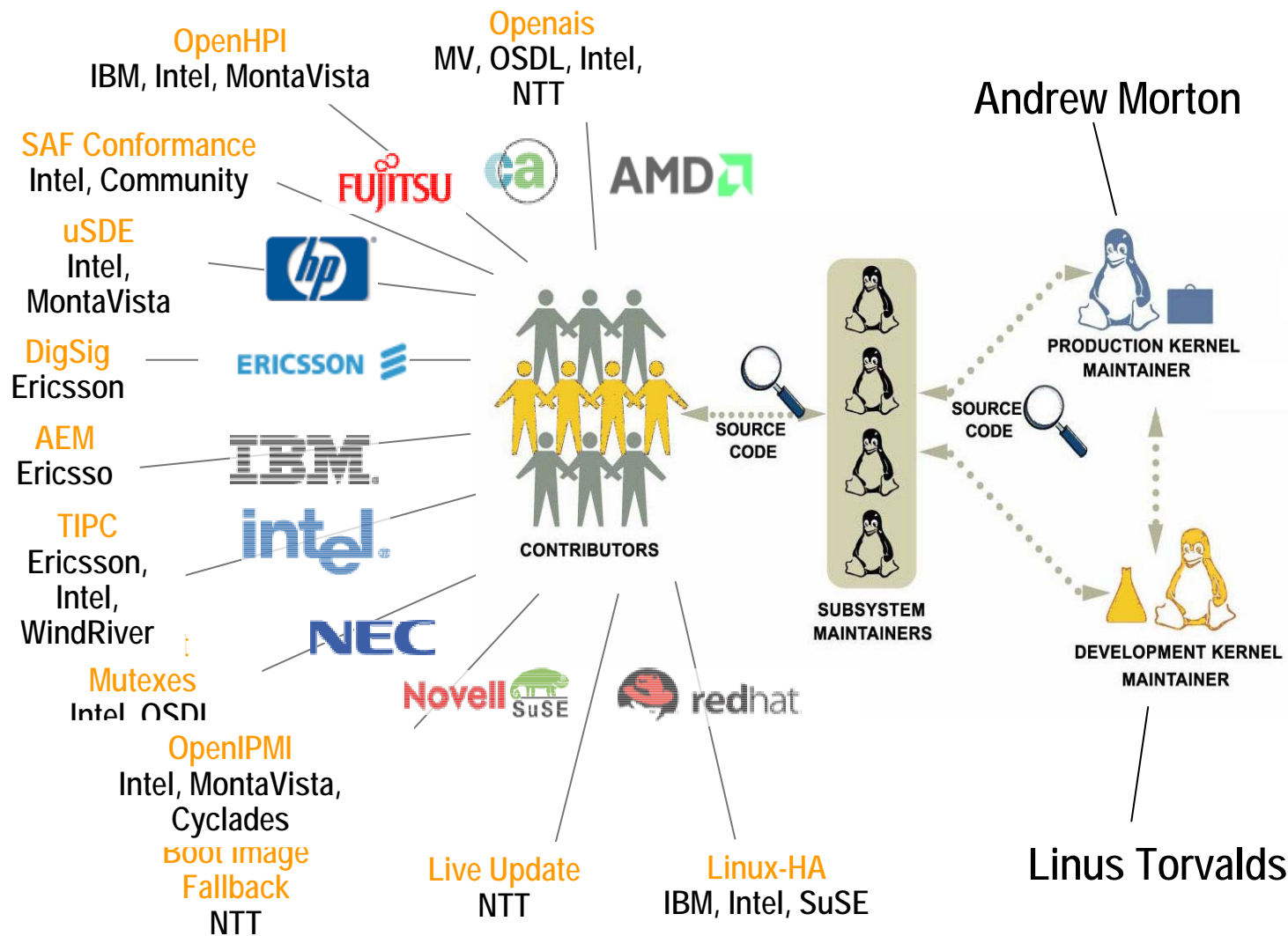
1. CGL specs calls out specific function needs in the Linux kernel – address usage models
2. Gap analysis is conducted to identify the current open source implementation gap against a given requirement
3. Member companies works with community and open source projects to fill the gap
4. As result, some of these features are mainlined



Mapping between requirements and code

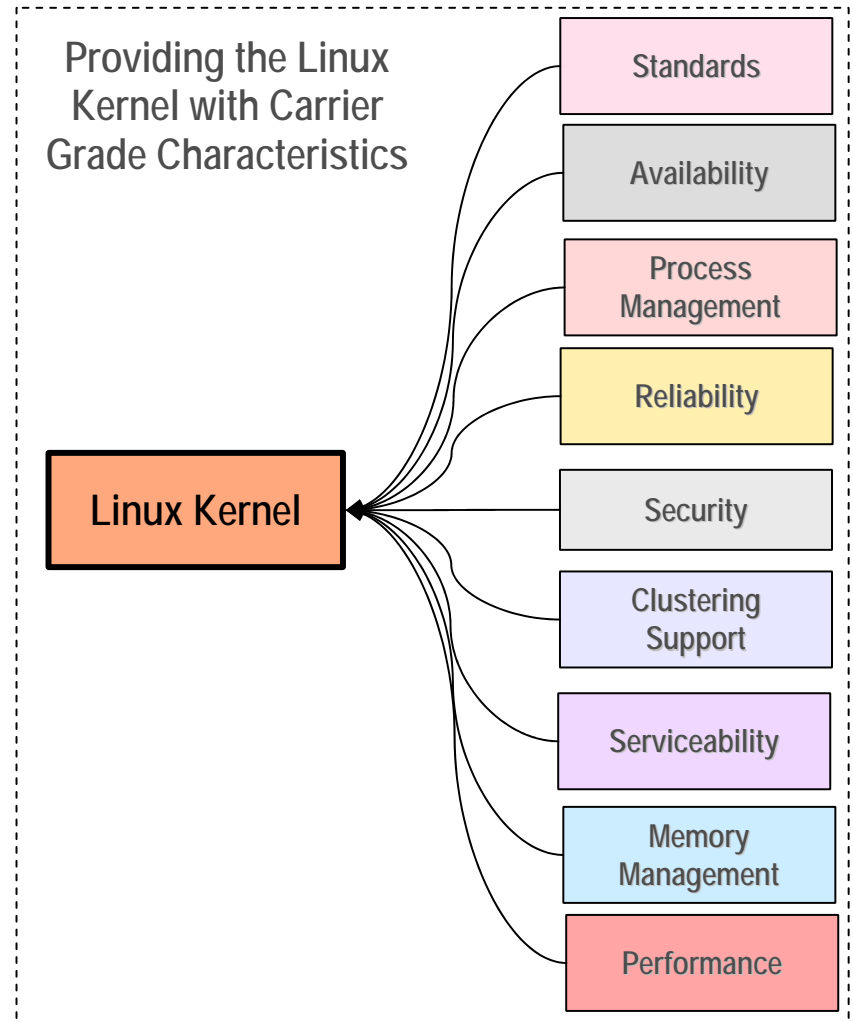
- There are many projects (with src code) that member companies are working on that are mapped to CGL's needs
 - Some projects carry the CGL flag
 - Others don't
- Sometime, it works better that way!
- At the end, as long as there is src code available in open source that fulfill CGL's need, we have achieved a milestone

CGL Member Participation in Development



Integration with Linux Kernel

- Kernel integration takes time
- Some enhancements already integrated with 2.6 kernel
 - Others will follow
- All enhancements available from SourceForge or project web sites



Forking?

- We shouldn't fear the fork
 - It is a normal part of the Linux development process
- There are short term forks that occur all of the time with the intention of being a proof of the technology before it is integrated into the mainstream.
- What is important?
 - The **purpose of the fork**
 - **With CGL, most of the times forks occur with the intention of proving the concept to the wider audience**
 - **Yes, sometimes that fork can last for over a year**
 - But **the goal is to work it back**
 - **It is a temporary thing not an idea to run off in a different direction**

A permanent patch?

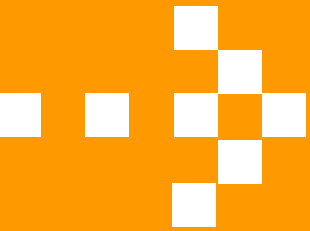
- Different than a fork
 - A way to get technology to track the mainstream that might be in opposition to the needs of the larger user population
- Linux tools support this environment and makes it easy for somebody to add in their little bit of required differences (based upon business needs) without incurring the expense of doing the whole stack
- Is this such a bad thing? Maybe not?
 - It allows for the special case where something that greatly benefits one industry segment but would hurt others can still track the normal development process and benefit from the larger amount of work being done.

Let's look at CGL 2.0 Requirement Document

- Total: 71 requirements
 - 18 in stock kernel
 - 6 in stock kernel + glibc
 - 7 in glibc
 - 8 in stock kernel + RPM (utilities)
 - 26 as RPMs
 - 5 as kernel patches
 - 1 unimplemented

What about the CGL 3.1 Requirements

- The focus now is on development efforts!

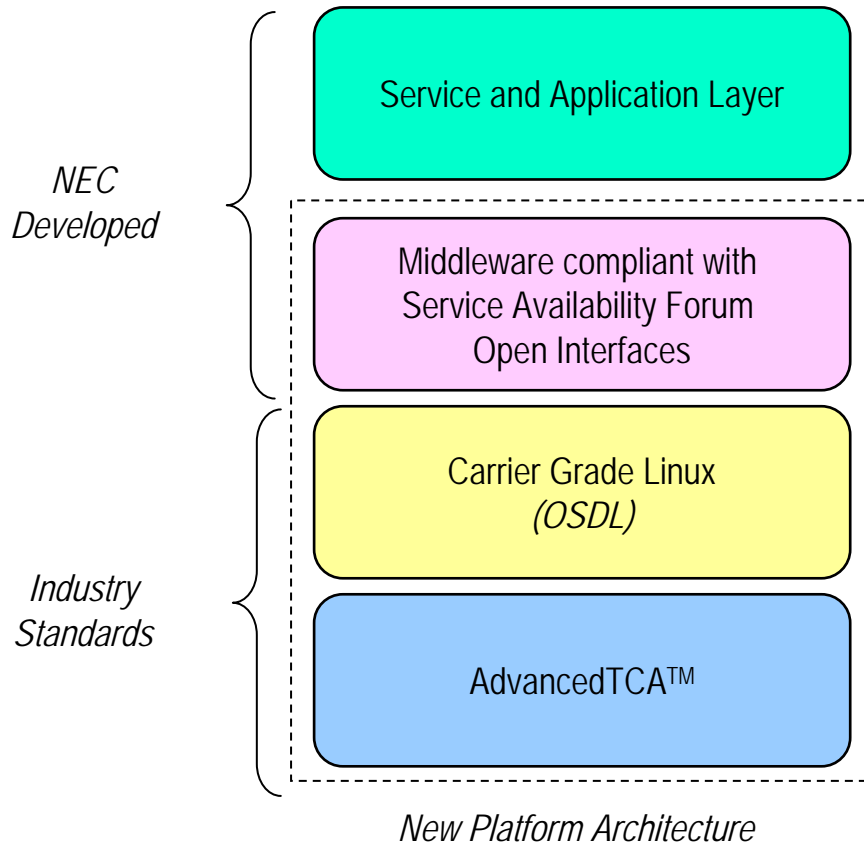


Carrier Grade Linux Deployments *(Linux domination in telecom networks)*

NEC Case Study

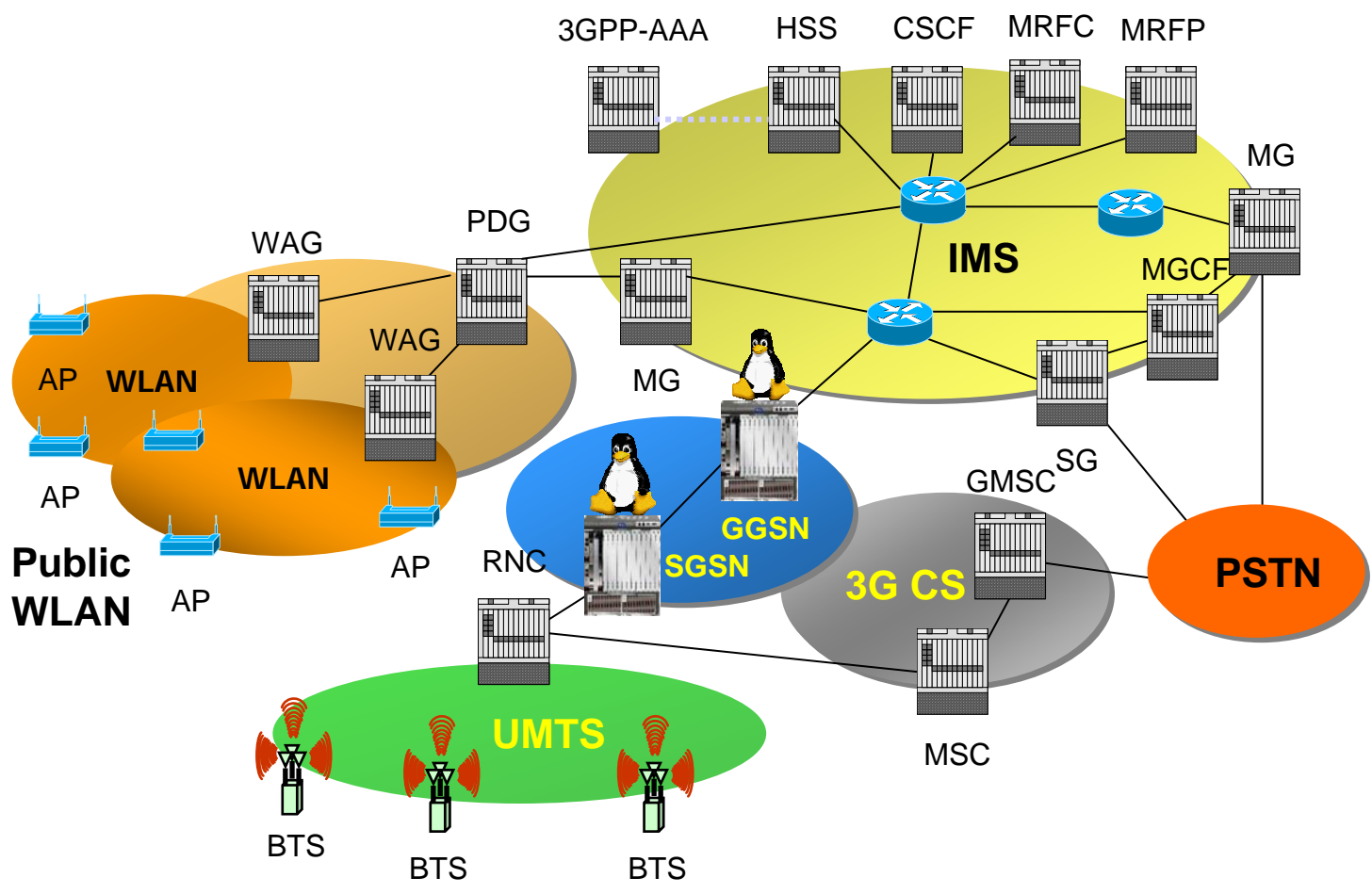
BT Case Study

NEC New Platform Architecture



Advanced New Platform for Mobile Operators Employing OSDL Carrier Grade Linux and PICMG Forum Advanced-TCA(TM)

Application to Mobile Infrastructure



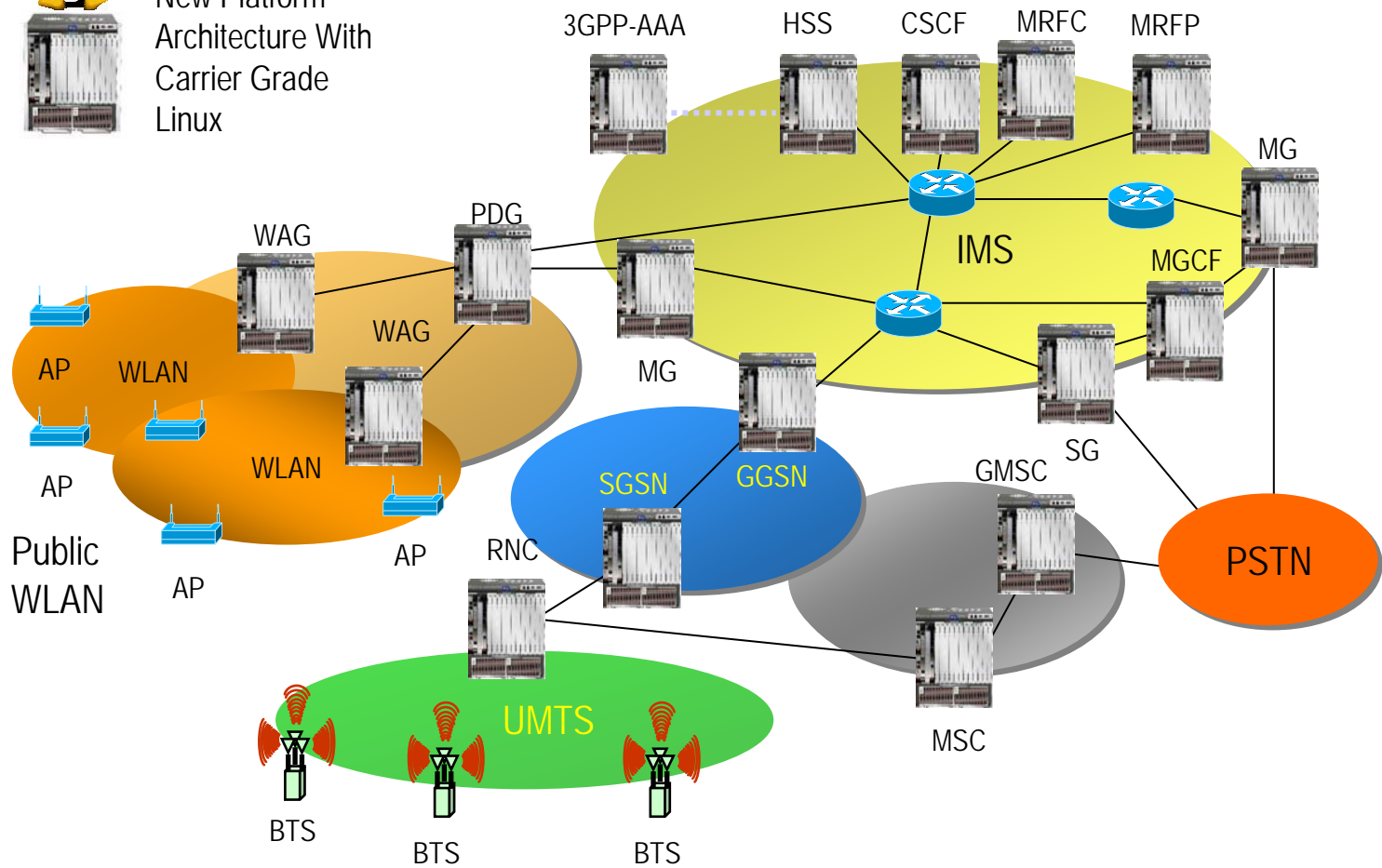
Current Status

- 100+ nodes in several carrier networks
- 10M+ subscribers in total
- Running for 6 months in actual commercial services

Future Planned Deployments

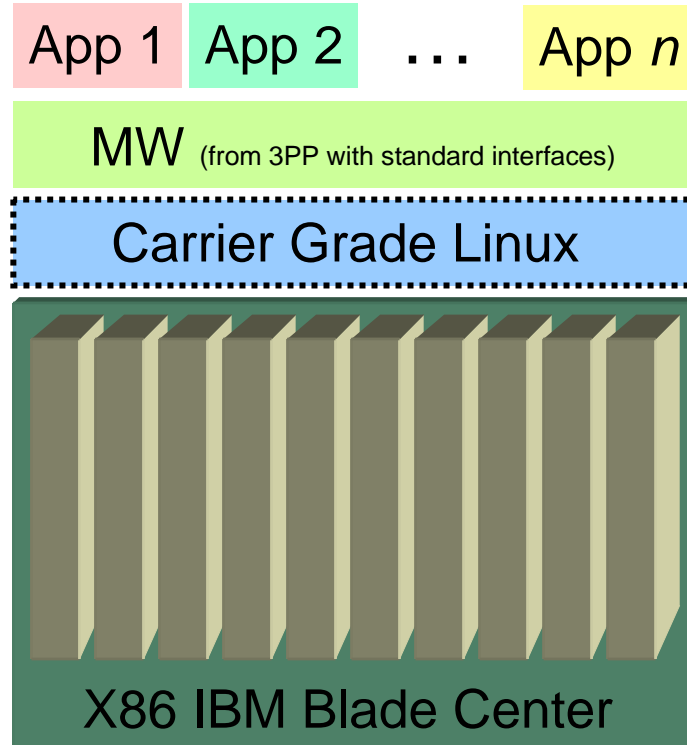


New Platform Architecture With Carrier Grade Linux

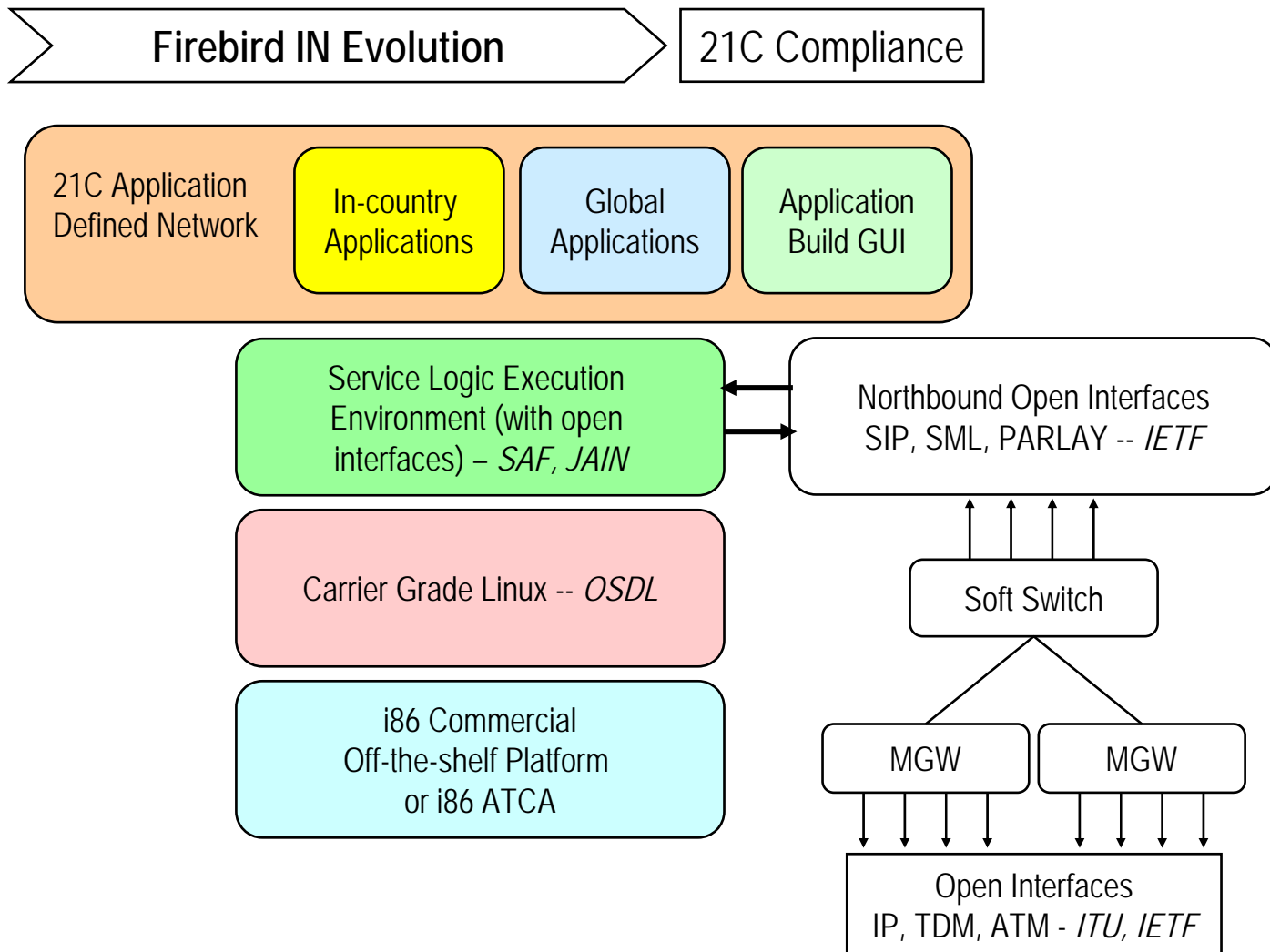


BT Global – Pilot Project in Belgium

- Provide voice communication services

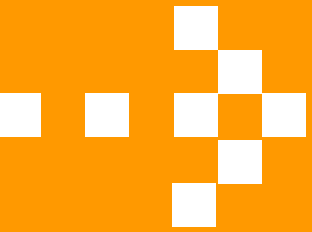


Firebird System Architecture



Results

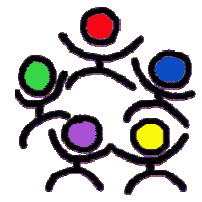
- CGL is suitable
- It has been a good experience
- Problems were fixed quickly
- Lower cost solution than proprietary solutions
- *Investigating roll out of the platform on all EU network*



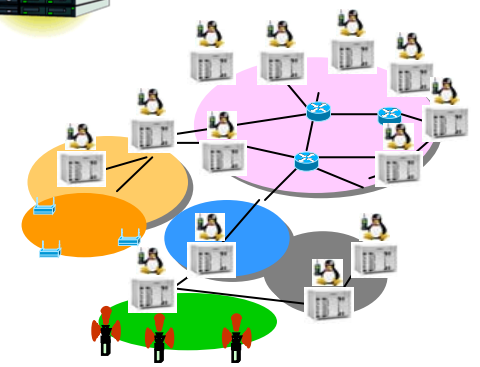
Conclusions

CGL Initiative: Jan 2002 – Present

- Increasing number of OSDL member companies involved with CGL
- 3 major releases of the requirement definition
- 8 Linux distributors are shipping CGL distributions, 5 of them have registered against CGL 2.0
- Over 20 providers for CGL based platforms



CGL 1.1 October 2002
CGL 2.0 October 2003
CGL 3.1 June 2005



Service Providers/Carriers are deploying CGL based platforms on their networks providing voice and data communications services to their subscribers



Thank You for the Linux Kernel community



**Helping us extend and
further the adoption of
Linux in Telecom!**