The MPPWG Mobile Phone API

Part I – The Working Group Process

CELF Embedded Linux Conference -- 12 April 2006

Itinerary

- **History and background for MPPWG**
- **Reference Tiers**
- **The Reference Architecture**
- **Specification Development Process**
- **Relationship to Other Organizations**

Rationale for Profile Work Groups

- **Need to better match the technology program to member companies' needs and capabilities**
  - CELF's work program had perviously been driven from OS technology perspective.
  - Many CELF members are primarily product builders and do not develop OS functionality, but use a distributor.
- **Product-side CELF members have the most direct perception of needs for supporting CE product development.**
- **Domain-centric working groups can be effective drivers for Linux technology for their specific domains.**

MPPWG Rationale (1)

- **The Mobile Phone domain is a prime domain for CELF – good match of membership with domain**
  - Most phone vendors are working on Linux-based products or prototypes.
  - Extensibility and availability of third-party software are increasingly important for phone vendors, especially at the high end.
  - Phone requirements stress Linux in all the areas CELF has identified as central to CE needs and others (like networking bandwidth).
  - The major phone vendors are participating in CELF.
  - Using mobile-phone requirements to identify technology needs common to many CE domains will ensure focus on activities of high value to many members.

MPPWG Rationale (2)

- **Many elements of common mobile-phone functionality are not yet available as commercially-qualified open-source components, including such central functionality as browsers.**

- **Many successful solutions to needs in the mobile-phone domain will also be applicable to other CE domains. The resulting common capabilities will avoid fragmentation and allow vendors to concentrate on differentiating features, rather than on base functionality.**

- **With a common framework defined, 3d-party and open-source projects will be able to develop new features knowing that they will plug into multiple vendors' profile-compliant products.**

Charter

- **The Mobile Phone Profile Working Group (MPP WG) will develop a reference profile for Linux-based mobile-phones in various functionality tiers.**
  - *Reference architectures* (components and basic structure) for platform and enabling services for *specific tiers*.
  - The reference architectures provide a framework for identification of *performance and functionality needs*:
    - Mobile-phone-specific requirements for the base kernel.
    - Mobile-phone-specific requirements for enabling middleware and services supporting horizontal (functionality) domains important in mobile phones (multimedia, Database, etc.).
  - A *roadmap* for the evolution of the profile, projecting need for additional component technologies.

Scope

- **The scope of the MPP WG includes Linux interfaces, middleware, APIs, and component implementations supporting phone-specific functionality tiers.**
  - User-level feature functionality and air-protocol support included only as sources of requirements and as plug-in points.
- **Where OSS components are not available, the profile may specify plug-in points, such as API definitions, where outside components can be connected to provide needed functionality.**

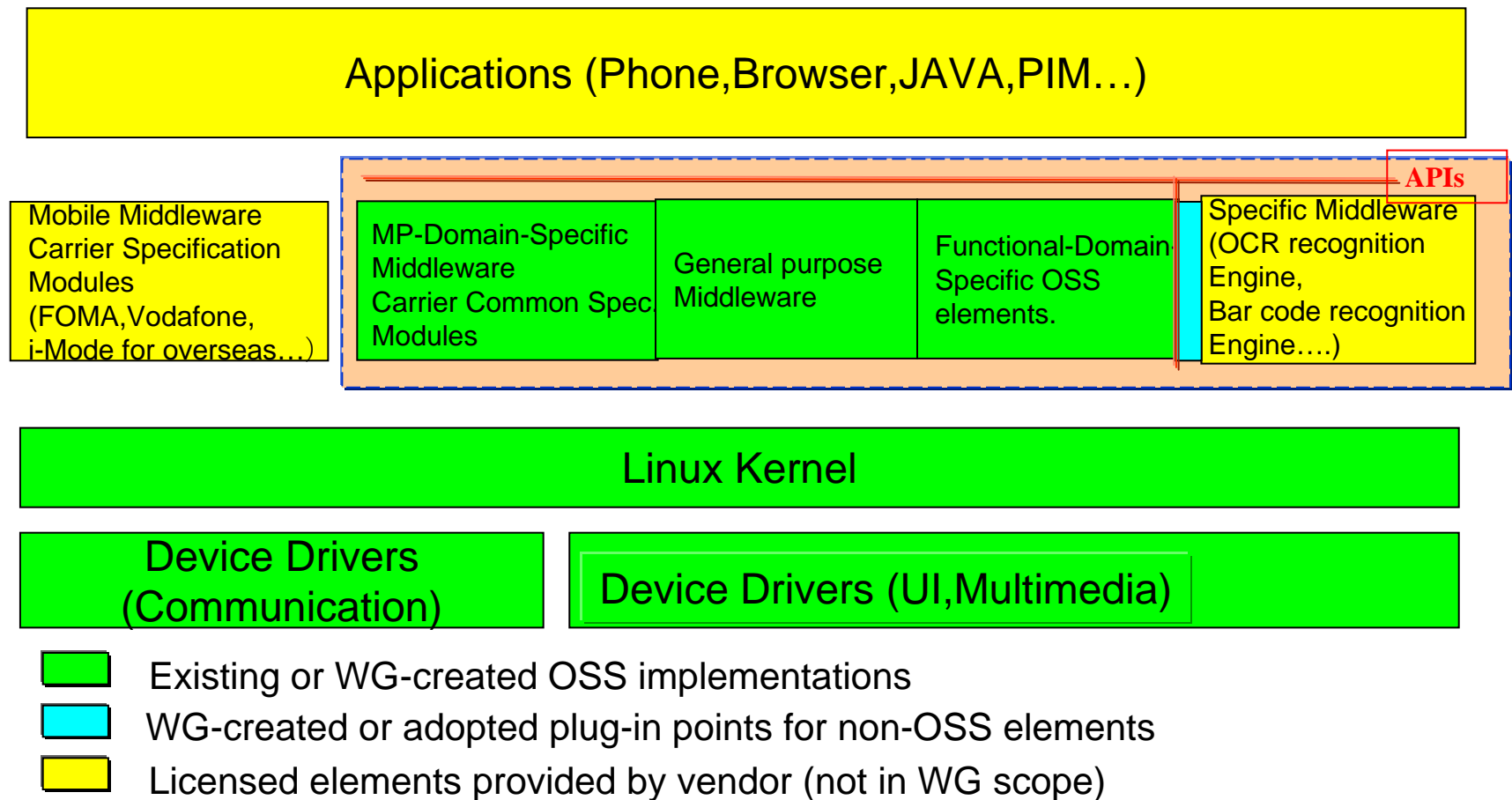# Deliverables

- **Requirements (to existing WGs and AG)**

- **Reference Tiers and Architecture(s) for Linux-based Mobile Phones (to AG for publication)**

- **Implementations of available Reference Architecture components (to AG for patch tree)**

Mobile-Phone Profile Reference Model

Applications (Phone,Browser,JAVA,PIM…)

APIs

Mobile Middleware
Carrier Specification
Modules
(FOMA,Vodafone,
i-Mode for overseas…)

MP-Domain-Specific
Middleware
Carrier Common Spec
Modules

General purpose
Middleware

Functional-Domain
Specific OSS
elements.

Specific Middleware
(OCR recognition
Engine,
Bar code recognition
Engine….)

Linux Kernel

**Device Drivers
(Communication)**

Device Drivers (UI,Multimedia)

Existing or WG-created OSS implementations

WG-created or adopted plug-in points for non-OSS elements

Licensed elements provided by vendor (not in WG scope)

Membership and Participation

- **Group Membership is broad**
  - 77 individuals on mailing list
  - 29 companies on mailing list
- **Three 2005 face-to-face meetings, one in 2006**
  - Meeting attendance has declined
- **Teleconference in January 2006**
- **First specifications now out for public Formal Review**

CE Linux Forum

Members

- **2WIRE**
- **Agere**
- **Aplix**
- **AXE**
- **ETRI**
- **France Telecom**
- **Freescale**
- **Fujitsu**
- **IBM**
- **Intel**
- **Kenwood**
- **LGE**
- **MontaVista**
- **Motorola**
- **Movial**

- **NEC**
- **Nokia**
- **Nvidia**
- **Palmsource**
- **Panasonic**
- **Philips**
- **Renesas**
- **Samsung**
- **Sharp**
- **Sony**
- **ST**
- **TI**
- **Toshiba**
- **Trolltech**
- **WindRiver**

Reference Tiers and Reference Architecture

Reference Tiers (1)

		Tier		
Aspect	Smart Phone	Media Terminal	Feature Phone	Plain-Old Mobile
Focus	Business focus	Personal/Entertainment Focus	Lifestyle Focus (voice plus social networking support features)	Voice
Primary Functionality	Full PDA functionality (Calendaring, address book)	Strong PIM support, personal content management features	Minimal PIM functionality (phonebook, datebook)	Phonebook and call logs
Extensibility	Extensible (downloadable features)	Limited extensibility (MIDlets or BREW)	Limited extensibility (MIDlets/BREW)	No extensibility
Multimedia	Optional	Vido capture support, Media/content players, stereo	Limited multimedia support (pictures, MP3, MIDI,Simple, low-frame-rate animations)	None
DRM	Optional	Multiple DRM schemes	Hard DRM (limits on copying any media of given types)	None
Camera	Optional	2-3 megapixel camera	VGA camera or no camera	No camera

Reference Tiers (2)

	Tier			
Aspect	**Smart Phone**	**Media Terminal**	**Feature Phone**	**Plain-Old Mobile**
Browser	XHTML Browser	XHTML Browser	WAP Browser (text-centric)	Embedded access to specific URLs
Display	QVGA or larger color display	QSIF or larger color display	QSIF or smaller color display	Small display (64x96), non-color
Interaction	Touchscreen UI or QWERTY keyboard plus pointing device	Specialized keypad for media/game interaction	Standard keypad plus carrier-specific keys	Standard keypad
Connectivity	3G connectivity, possibly WLAN, Bluetooth, IrDA	2.5G or 3G connectivity, possibly WLAN; High-speed USB; Bluetooth	2G connectivity; USB or serial cable	2G connectivity; proprietary accessory cable
Memory	32M RAM, 64M ROM, removable storage	64M RAM, 64M ROM, Hard Disk or large removable storage	16M RAM, 16M ROM, no removable storage	8M RAM, 8M ROM or less
Processor	120MHz	200MHz	30MHz	15MHz

CE Linux Forum

The Reference Architecture

CE Linux Forum

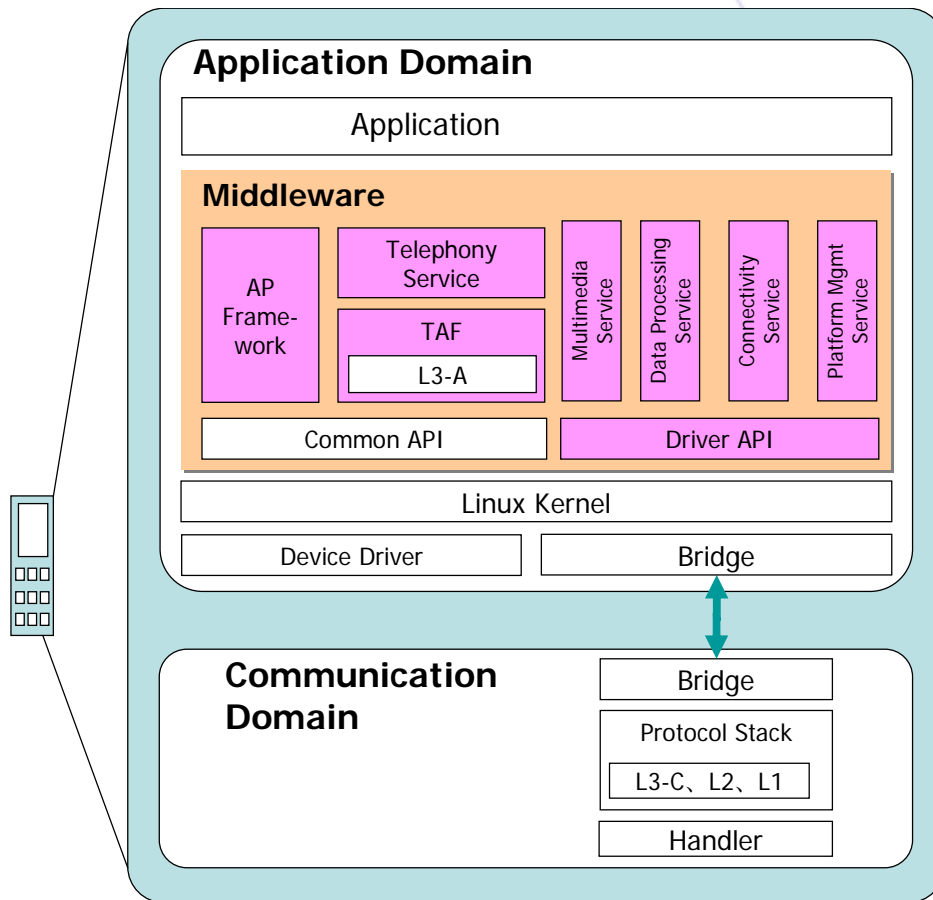# The Reference Architecture Specification

- **A reference architecture is a commonly understood approach to structuring a kind of system**

- **Intended to lay out the parts of the system so that the implementation and APIs can be mapped to reference components**

- **Helps clarify what is in-scope**

- **The Reference Architecture is non-normative – it is meant to provide background and understanding, not to constrain conforming implementations of the APIs**

Overall Architecture

Application Domain

Application

Middleware

AP Frame-work	Telephony Service	Multimedia Service	Data Processing Service	Connectivity Service	Platform Mgmt Service
	TAF				
	L3-A				

Common API	Driver API

Linux Kernel

Device Driver	Bridge

Communication Domain

Bridge

Protocol Stack

L3-C、L2、L1

Handler

- Two domains may be separate processors or share one

- App domain has four layers:
  - Applications
  - Middleware
  - Kernel
  - Drivers and Bridge

- Communication Domain includes air stack and real-time activities
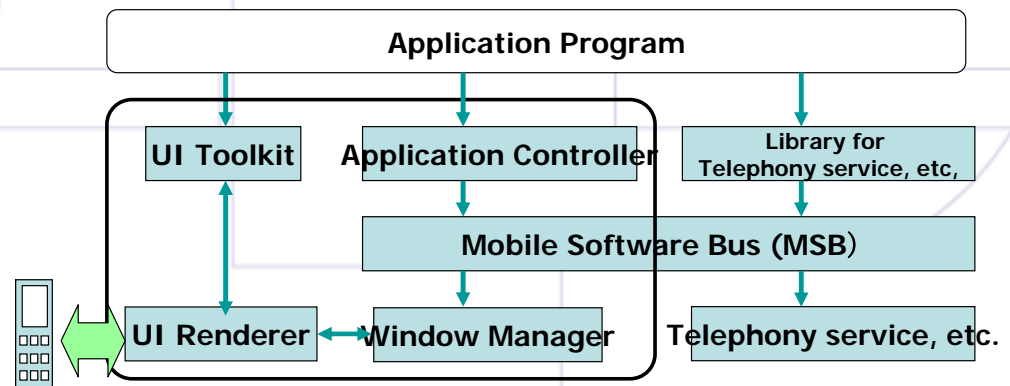
Kernel

- **Kernel is standard Linux with embedded patches**
    - Boot-time improvements
    - System size reduction
    - Dynamic power management
    - Compressed filesystems
    - Security enhancements
- **Real-time support or microkernel may be required if running on single processor**
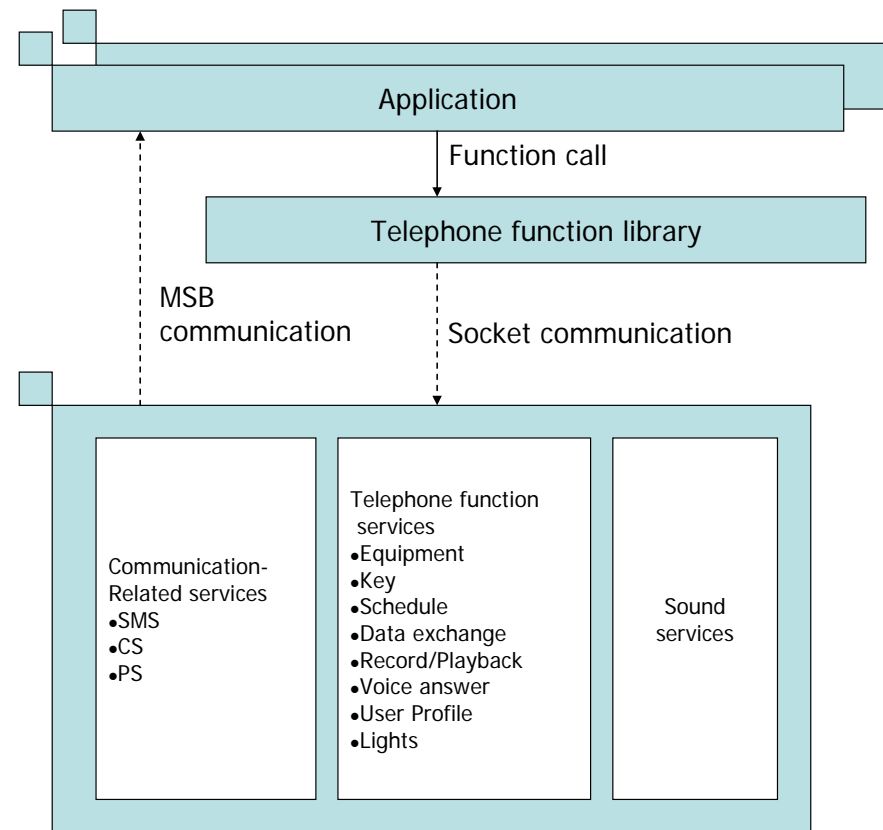
Application Framework

- **Provides both application management (launch and communicate) and UI framework**

- **Reference Architecture does not specify toolkit or rendering components**

- **MSB provides "event bus"**
  - Registration and filtering
  - Event delivery in client process

Application Program

| UI Toolkit | Application Controller | Library for Telephony service, etc, |

Mobile Software Bus (MSB)

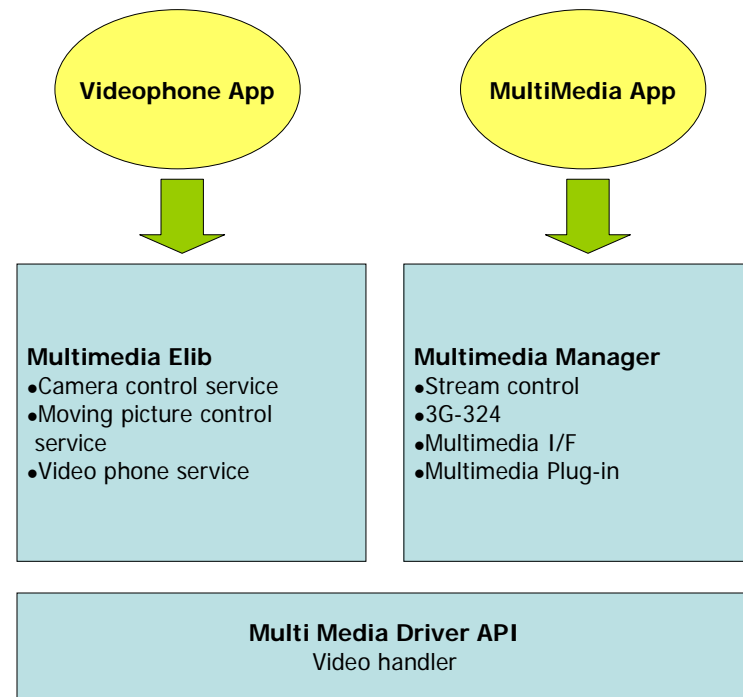| UI Renderer | Window Manager | Telephony service, etc. |

Telephony Framework

- **Provides various kinds of communication services**
  - Circuit-switched (voice
  - Pack-switched (data)
  - SMS
- **Also includes handset con functions**
  - Lights
  - Ringers
  - Audio control

Application

Function call

Telephone function library

MSB communication

Socket communication

Communication-Related services •SMS •CS •PS	Telephone function services •Equipment •Key •Schedule •Data exchange •Record/Playback •Voice answer •User Profile •Lights	Sound services

Multimedia Framework

- **Video telephony support**
- **Audio/video stream management**
- **Multimedia plugins and rendering**
- **Multimedia drivers**

Videophone App

MultiMedia App

Multimedia Elib
- Camera control service
- Moving picture control service
- Video phone service

Multimedia Manager
- Stream control
- 3G-324
- Multimedia I/F
- Multimedia Plug-in

Multi Media Driver API
Video handler

CE Linux Forum

API Specification Development

Sources

- **Working group polled members for reference architecture proposals and contributions of APIs and implementations**

- **Reference Architecture and API submitted by NEC and Panasonic**

  – Based on working implementation for DoCoMo FOMA phones

Specification Development Process

- **Draft Development Phase**
  - Editors prepare drafts, respond to review comments
  - Face-to-face and e-mail reviews
  - WG polled for approval as review candidate
  - Must have at least 5 supporting members
- **Formal Review Phase**
  - Available to interested parties
  - All comments must be resolved or rejected
  - Approval vote at editor's request
  - If approved by WG vote, published as Proposed Specification
- **AG Review Phase**
  - All AG issues resolved
- **Board of Directors Review Phase**
  - All Board issues resolved
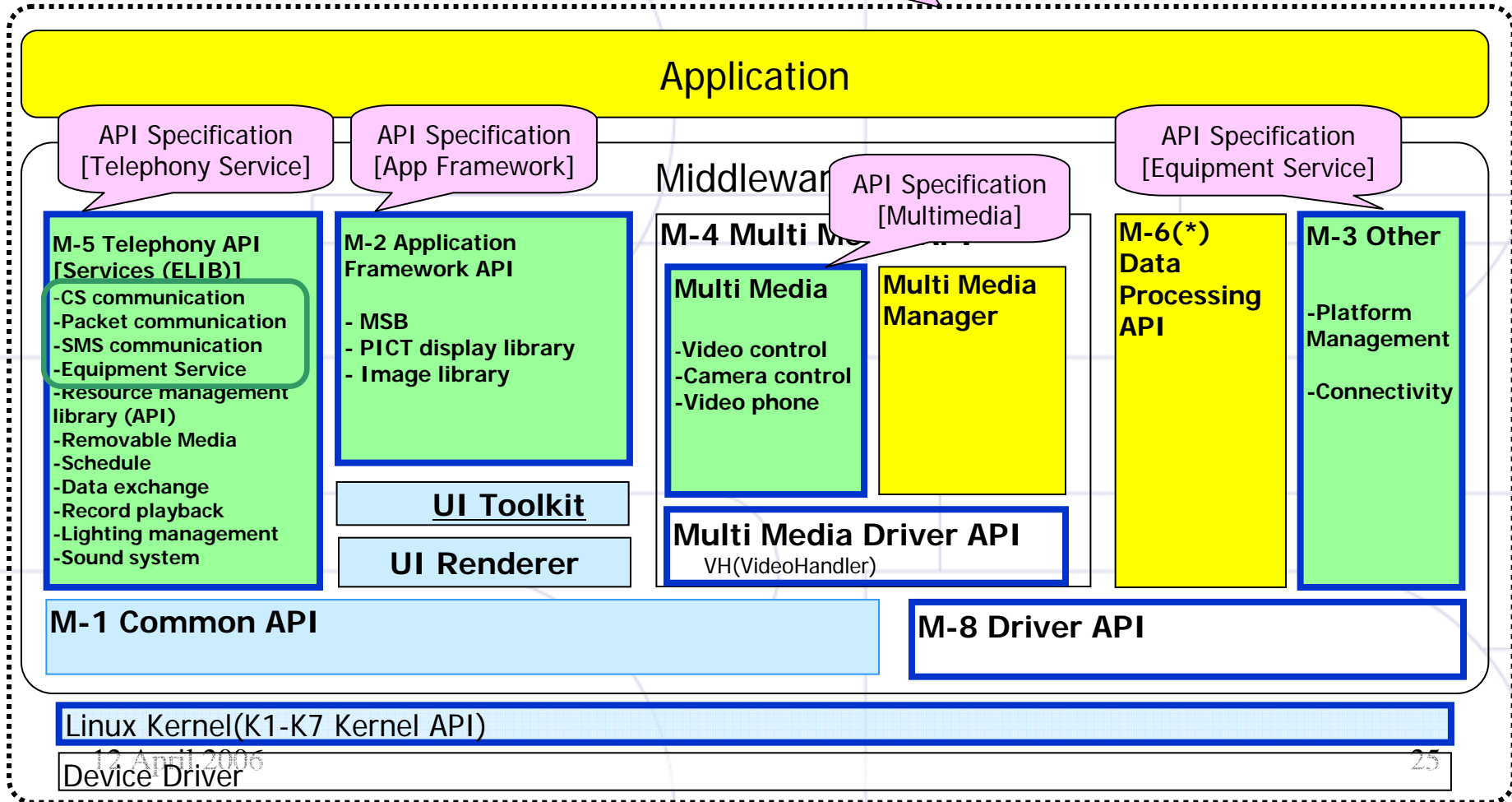 - Publication as Approved Specification

API Specifications

Reference Architecture Specification

···· : Reference Architecture

▭ : API Function List

▬ : API Specification

Application

API Specification [Telephony Service]

API Specification [App Framework]

Middleware

API Specification [Multimedia]

API Specification [Equipment Service]

M-5 Telephony API [Services (ELIB)]
- CS communication
- Packet communication
- SMS communication
- Equipment Service
- Resource management library (API)
- Removable Media
- Schedule
- Data exchange
- Record playback
- Lighting management
- Sound system

M-2 Application Framework API

- MSB
- PICT display library
- Image library

M-4 Multi Media API

Multi Media
- Video control
- Camera control
- Video phone

Multi Media Manager

M-6(*) Data Processing API

M-3 Other
- Platform Management
- Connectivity

UI Toolkit

UI Renderer

Multi Media Driver API
VH(VideoHandler)

M-1 Common API

M-8 Driver API

Linux Kernel(K1-K7 Kernel API)

Device Driver

Specifications in Formal Review

- **Four documents released for Formal Review:**
  - Reference Architecture (Motorola)
  - Preface (Aplix)
  - Circuit-switched Communications (Aplix)
  - Packet-switched Communications (NEC)
- **Reviewer sign-up and comments submission via the MPPCOMMENTS mailing list at list.celinuxforum.org**

Specifications in Development

- **Partially edited and awaiting review**
  - Short Message Service (MontaVista)
  - Equipment Services (ETRI)
  - Lighting Services (NEC)
- **Supporting non-Specification (not advancing)**
  - Programming Guide
- **Not yet underway**
  - Six additional Telephony sections, Multimedia, Other Services, and Application Framework API

Formal Review Process

- **Issue call for reviewers**
  - Invite CELF members, LiPS Forum, MLI, others
  - Public announcement on home page calling for reviewers
- **Review period runs 1 March – 23 April**
  - Reviewers register with MPPWG and submit comments
  - Reviewers are individuals not companies
- **Comments resolution period runs as long as necessary**
  - Editor can request vote when comments all resolved or rejected
- **WG votes on approval at request of Editor**
  - One vote per WG-member
  - Two-week e-mail voting period
  - Approval requires at least 5 supporting votes, and at least 75% of votes cast
- **If approved, publish as Proposed Specification and send to AG**

Comment Submission

- **Reviewers send registration and comments to:**

  MppComments@list.celinuxforum.org

- **Comments submitted in template spreadsheet or XML template**

- **Document Editor incorporates submissions into master tracking sheet**

- **Spreadsheet will be attached to MppApiIssues page in Public wiki**

  – Tracking spreadsheet is visible to reviewers and WG

Comment Resolution

- **Document editors will manage issue resolution for their chapters**
  - Editor tracks comment responses in comments spreadsheet
    - Enters proposed response in spreadsheet (accept fix, apply different fix, reject issue)
    - Leads review by WG and comment submitters
  - Editor can make non-substantive (language, etc.) changes to improve document *ad lib*
- **All comments must be addressed (accepted or rejected)**
- **Editor must verify resolutions with reviewers who raised them**
- **Editor periodically prepares updated drafts with list of CRs resolved**

Reference Implementation

- **Strong preference by Architecture Group for having a working implementation before approval**

- **Alternatives include:**

  – Form an OSS project to build a reference implementation

  – Cooperate with outside party

  – Refer to existing commercial offering as proof-of-feasibility

  – Could be built on top of some existing reference stack

![CE Linux Forum]

Other Organizations

LiPS Forum

- **Linux Phone Specification Forum launched in November**

  – Carrier-centric mission to produce consistent API across handset and "converged device" manufacturers to enable carrier customization

- **Has drafts of reference model and is working on event model**

- **Scope is similar to MPPWG (middleware/application services API) plus broader support program (conformance tests, etc.)**

- **Some cross-membership with CELF**

OSDL Mobile Linux Initiative

- **OSDL has chartered an initiative to promote Linux improvements for mobile-device applications**

- **Scope is similar to the other (non-MPPWG) parts of CELF**

- **Significant cross-membership with CELF and with LiPS Forum**

- **MLI is still working out the details of its work program and goals; focus is narrowly mobile, rather than all of CE**

LiPS Forum Interaction

- **Working with LiPS Forum to set up a liaison relationship, to avoid fragmentation**

  - MPPWG chair has attended LiPS technical meetings as observer

  - LiPS has supplied working documents (reference model) and comments on draft of specifications

- **LiPS Architecture WG Leader is on our mailing list as invited member of the MPPWG**

  - Met with us at February face-to-face

- **Planning to meet jointly in May, at OSDL face-to-face**

OSDL Mobile Linux Initiative Interaction

- **MPPWG Chair participating in bi-monthly teleconferences with MLI**

- **MPPWG participated in a joint face-to-face meeting hosted by the MLI in February; will also participate in May face-to-face**

  – LiPS Forum also participant in these events

CE Linux Forum

Comparative Scope (Source: LiPS)

	LiPS	CELF / MPP WG	OSDL
Organisation Description / Objectives	**Consistent deployment of UX, apps and services across all phones**	**Consistent services base for application deployment across mobile phones**	**Promotion of Linux in all commercial environments (servers to terminals)**
Phone Focus	**All telecom terminals (Fixed,Mobile,Converged)**	**Mobile Only** (**Other parts of CELF cover other devices**)	**N/A**
Driver	**Telecom Operators**	**H/W Manufacturers**	**Linux Community**
Deliverables	- **Requirements by profile** - **Application & Service enablers** - **User experience** - **Tools** - **Testing & certification**	- **Reference Architecture** - **Middleware APIs** - **Vertical approach** - **Combination of low and high level components**	- **Kernel optimisation** - **Silicon integration** - **OS Services** - **Testing and certification**

LiPS/CELF Overlap (Source: LiPS Forum)