2013 17th International Conference on Intelligence in Next Generation Networks
icin 2013
Unlocking Value from the Networks
Smarter Networks for Smarter Things
15-16 October 2013, Venice, Italy

WebRTC

Catalysing the Success of WebRTC for the Provision of Advanced Multimedia Real-Time Communication Services

Luis López, Miguel París, Raúl Benítez
(Universidad Rey Juan Carlos)

Javier López, José Antonio Santos
(Naevatec)

Universidad Rey Juan Carlos

NaevA Tec

fi-ware

Title

Part I
WebRTC technologies: current state of the art

Part II
WebRTC technologies: current problems and future trends

Part III
Kurento: a media server architecture for Future Internet RTC

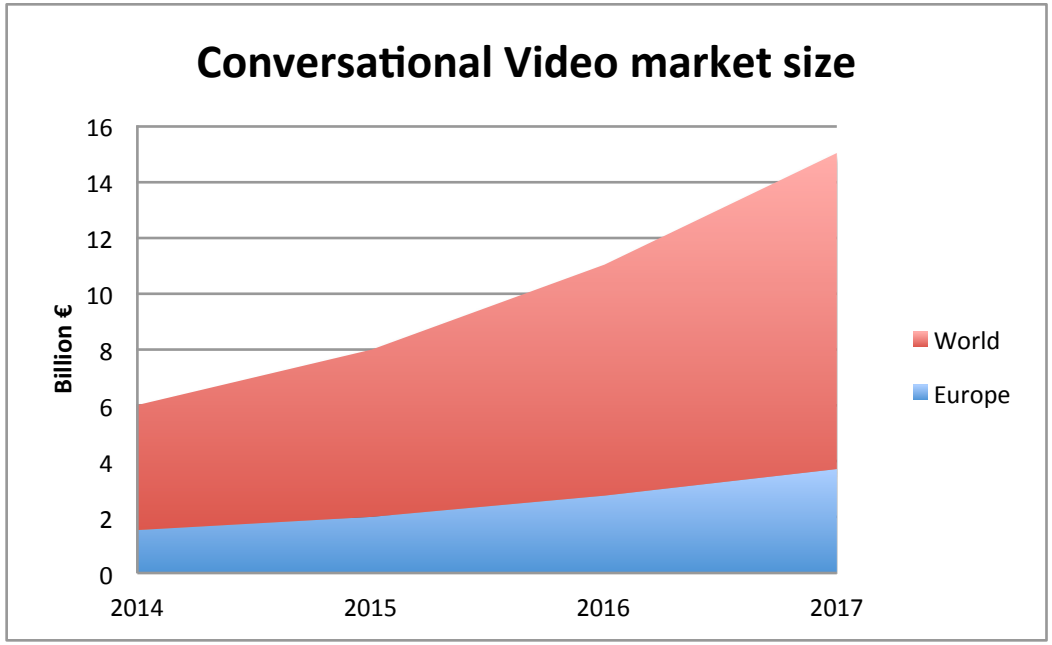
Part IV
Media API concepts

Part V
Developing WebRTC applications with Kurento

Part VI
Conclusion: present and future of the Kurento project

Real Time Communications

Global opportunity for video enabled RTC services and technologies

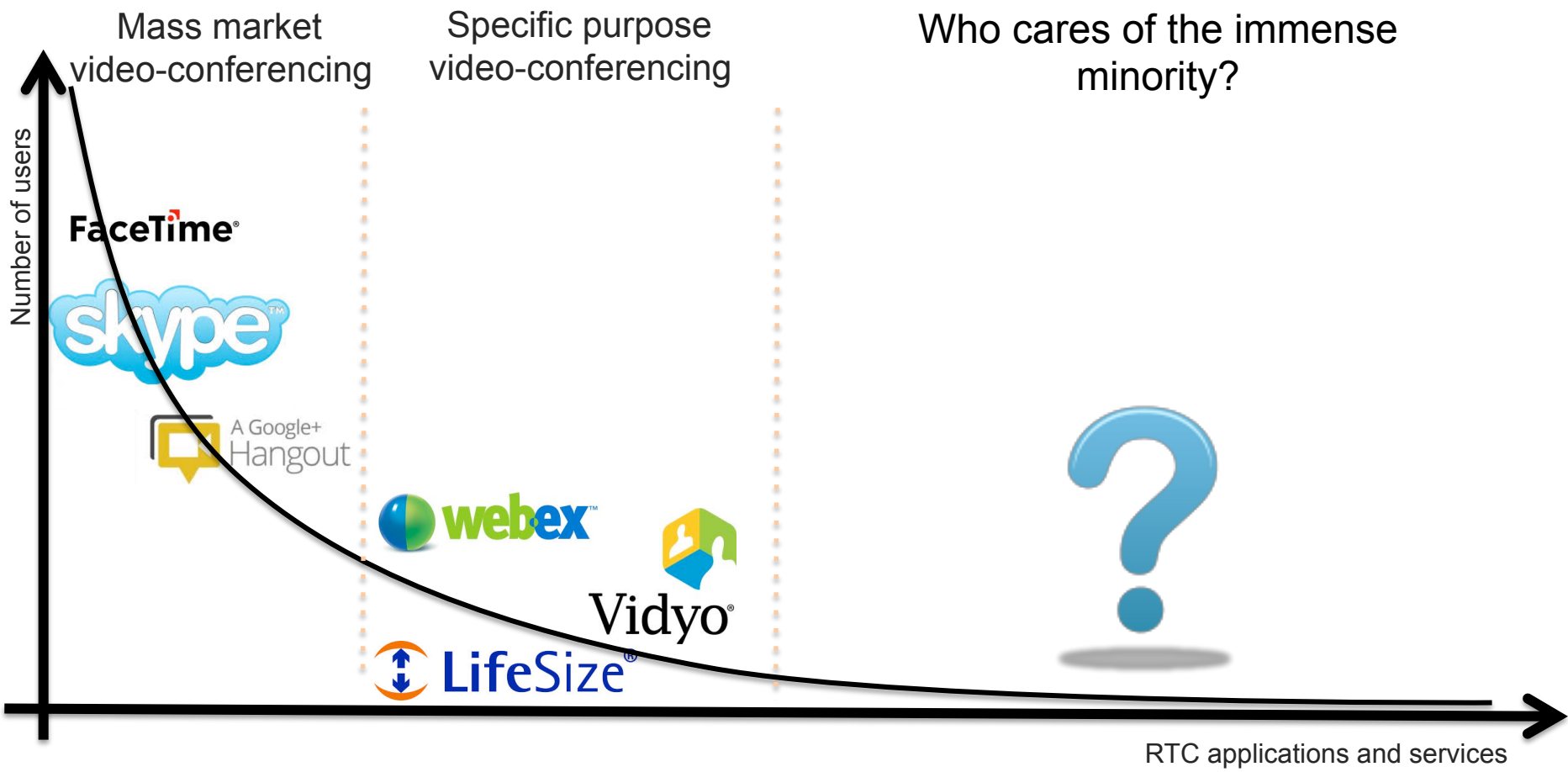


*Global Industry Analysts Inc.

RTC Video Services: Problem I

Fragmentation of services

RTC Video Services: Problem II



Disrupting the RTC market

WebRTC: a definition

- Framework, protocol and API that provide real-time voice, video and data in web browsers and other applications (by Salvatore Loreto)

WebRTC as an API

- Capabilities are exposed to web developers in an abstract manner and adapting to HTML5 philosophy
- Basing on standards: WebRTC Working Group of the W3C for JavaScript APIs

WebRTC as a framework

- Technological capabilities enabling RTC on web browsers: Codecs, NAT traversal, security, transports, etc.
- Basing on standards: RTCWeb Working Group of the IETF for protocols

Part II: WebRTC lacks & trends

WebRTC if fine for …

- P2P RTC between compatible browsers, but beyond that … *there be dragons*

.. but lacks: group communications

- No standardized mechanism for supporting group communications
- Only solutions based on simple "room" models (no push)

… but lacks: interoperability with legacy infrastructures

- No standardized WebRTC media gateway/middleware
- Limited interoperability capabilities

… but lacks: value added multimedia capabilities

- Media recording, media playing … are they possible beyond the client?
- Future Internet capabilities: Computer Vision, Augmented Reality … huge integration efforts

WebRTC risk: again fragmentation

Services are growing as isolated "cloud" islands	Tokbox, Lynckia, Crocodile, Bistri, …
	… provide group communications and recording but
WebRTC does not standardize a signaling plane	SIP, XMPP, REST, …
	… no simple ways of making them to interoperate
Commercial pressures	Microsoft is pushing alternative standards …
	… what about telcos?
Legacy technologies	Flash, silverlight …
	… a fraction of browsers only support those

Part III: Defeating fragmentation and catalyzing the long tail

Defeat fragmentation through an open philosophy

- Open standards
- Open source

Simplify development process and developer work

- Developers are the cornerstone of successful long-tail services
- No more RTC development nightmare

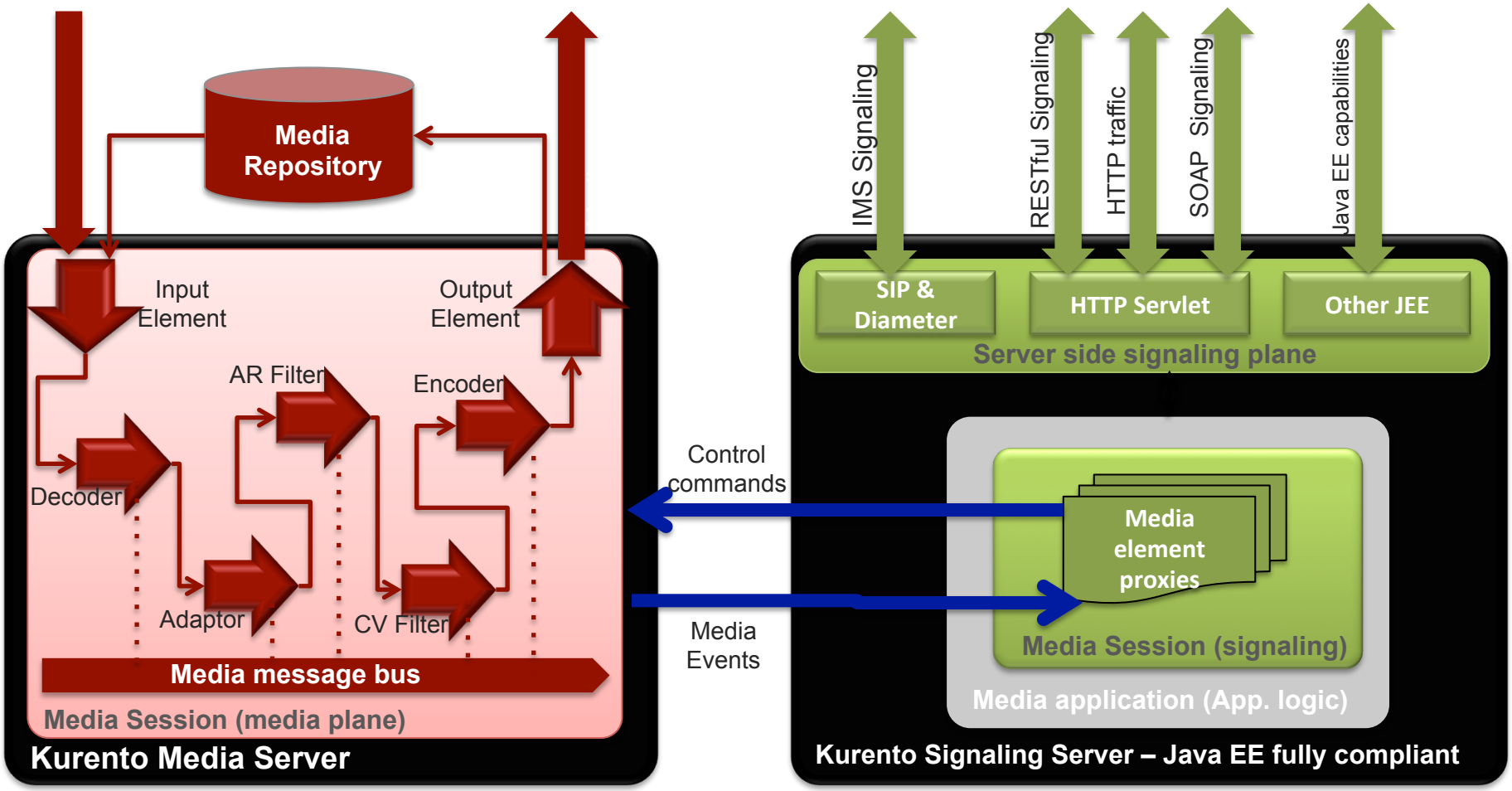
Consider the network side and not only the client side

- WebRTC fragmentation is coming from the network side of the service
- Need to consider “rare” network side services … IoT, CV, AR

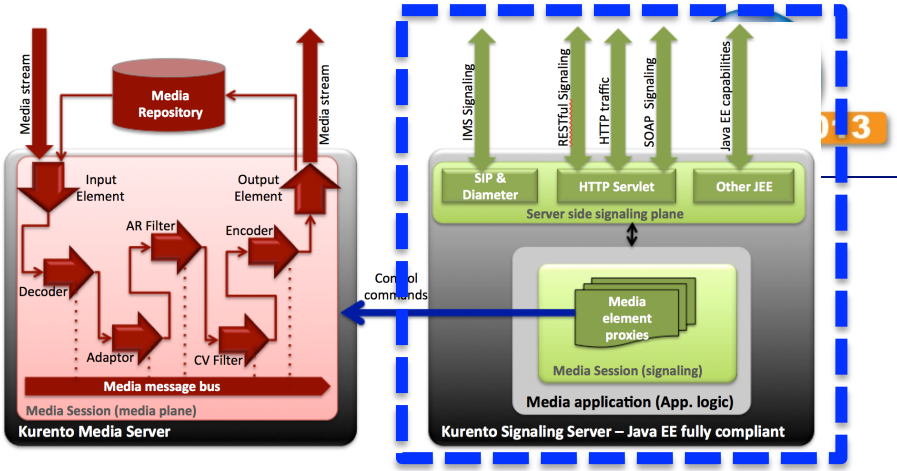
Don’t impose … reinforce

- Don’t impose a specific signaling plane
- … but make really simple to base applications on a given (standard) one
- .. but make simple to interoperate different signaling schemes

Kurento architecture

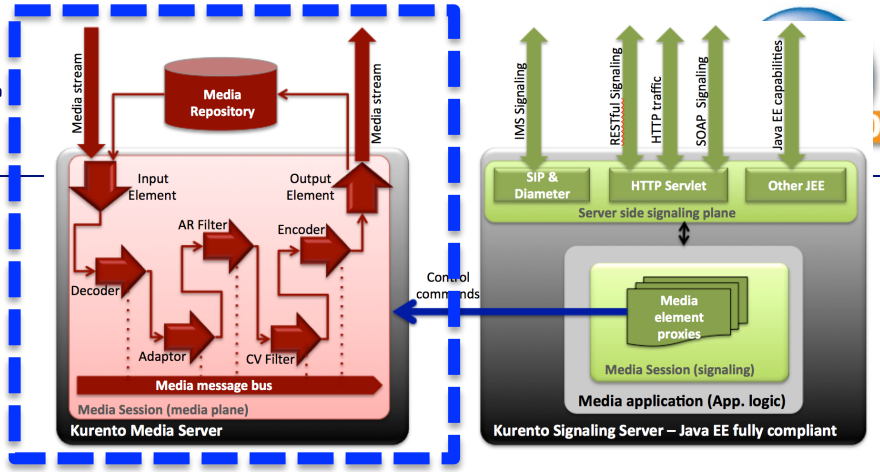


Kurento App. Server



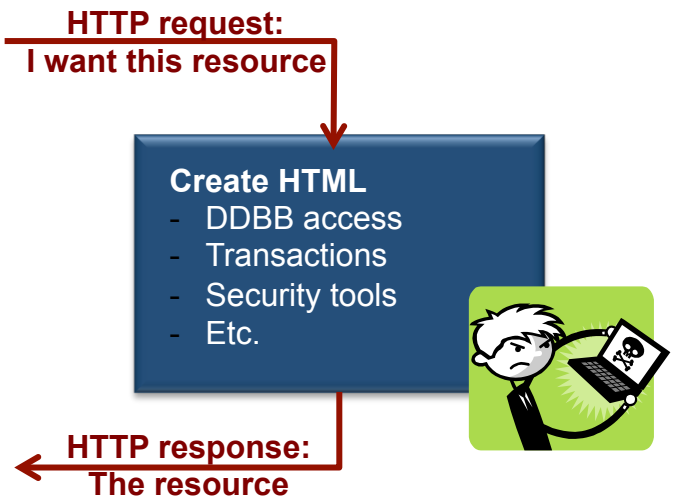
Based on JBoss Java EE microcontainer	All Java EE capabilities are available (long tail)
	Provides built-in signaling APIs (SIP&HTTP Servlets)
APIs are created on top of it	App. developers only interact with the AS
	Fast development based on well-known Java technologies
Seamless WWW & IMS integration	JBoss & Spring WWW development capabilities available
	Mobicents extensions (Diameter, SS7 GW) for IMS/Telco
Open Source	LGPL
	Very active community

Kurento Media Server

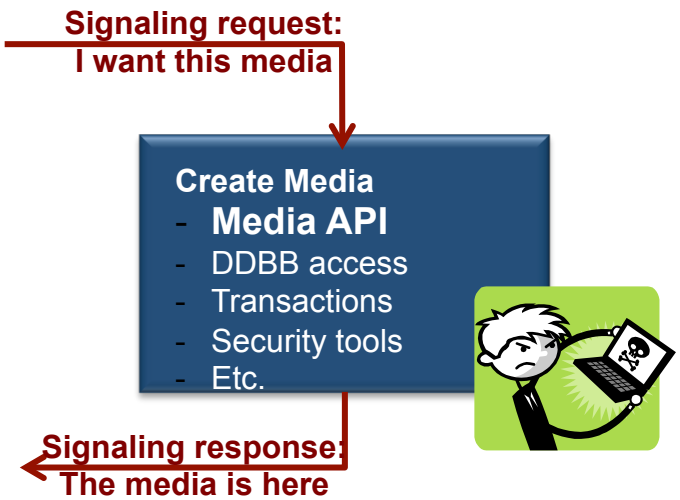


Base on GStreamer	Support for the most popular codecs and formats
	Support for common RTC transports
Fully abstracted	App. developers are not in contact with it
	Low level complexities are abstracted
Proxies and RPCs	Media capabilities are controlled through proxies in the AS
	Thrift RPCs are used for controlling the MS from the AS
Open Source	LGPL
	Very active community

A WWW inspired API



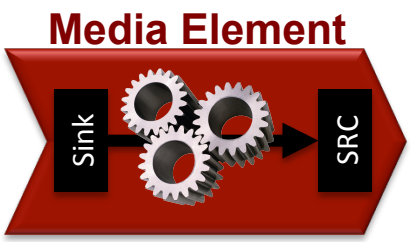
Intuition behind a traditional WWW application



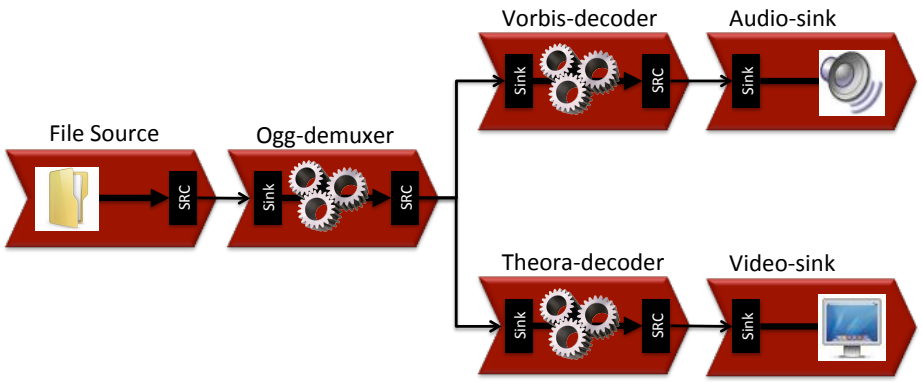
Intuition behind Kurento development APIs

Part IV: Core Media API concepts

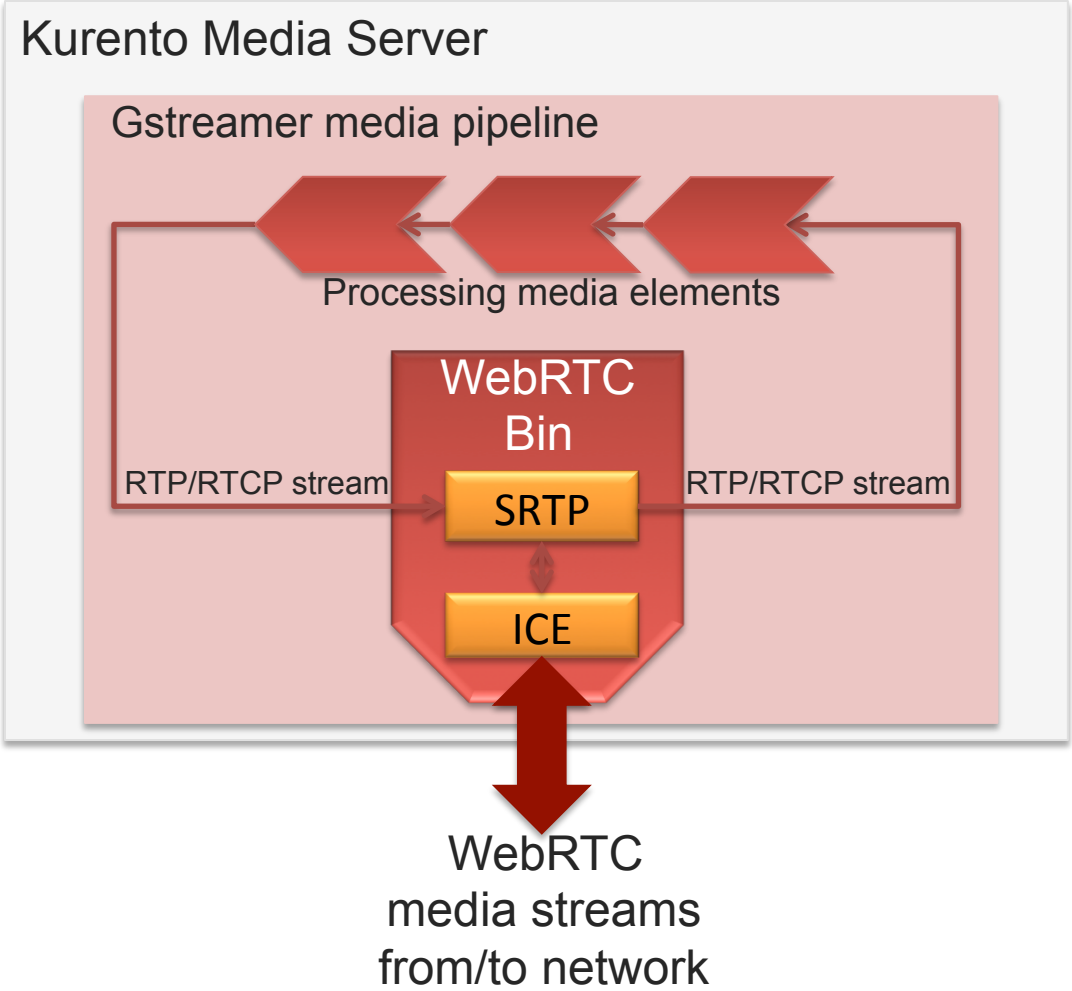
- Media Element
  - Provides a specific media functionality
    - Building block
    - Send/receive media
    - Process media
    - Transform media
  - The Media API provides a toolbox of media elements ready to be used.
  - New media elements can be added



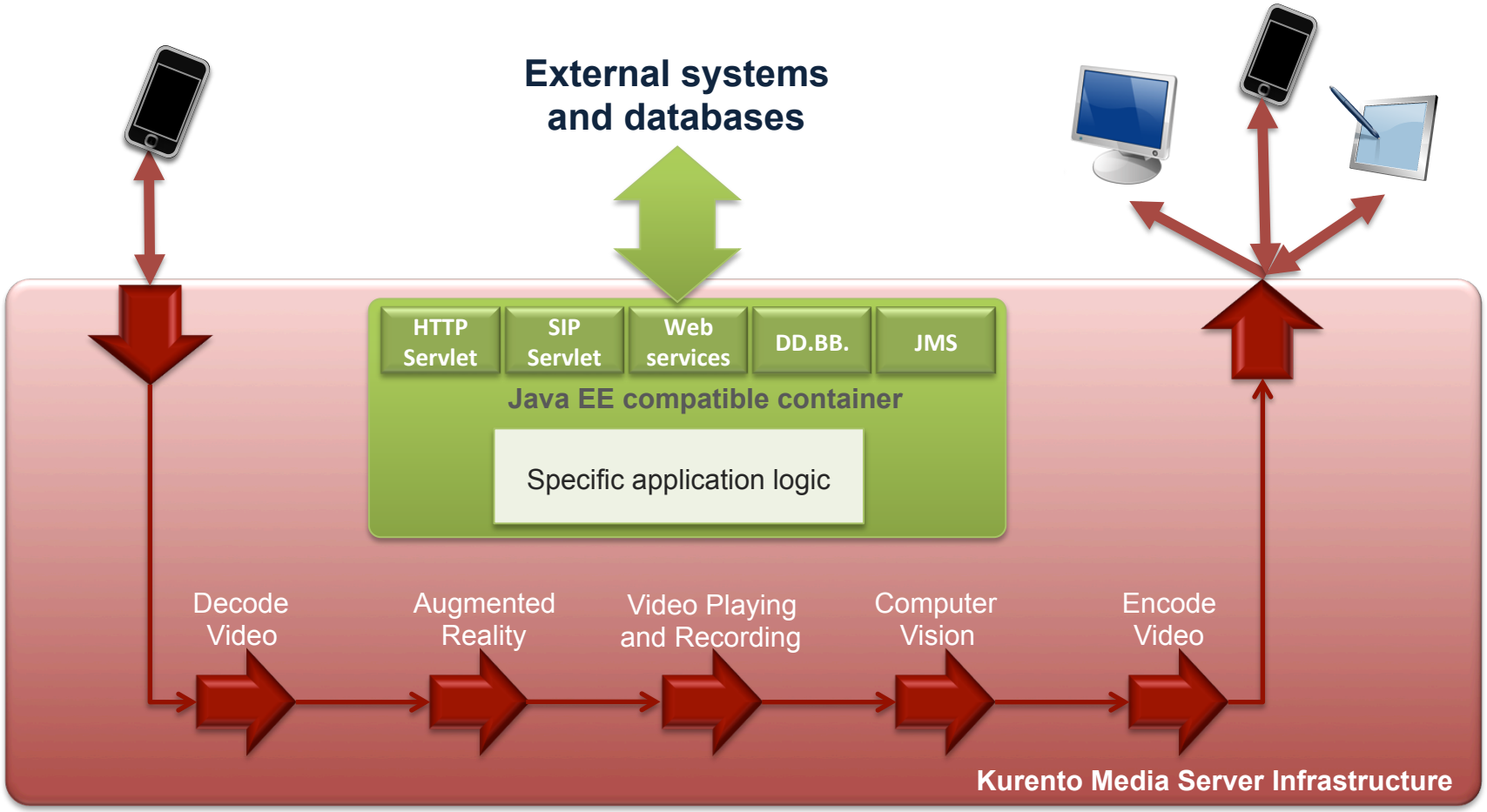
- Media pipeline
  - Chain of media elements implementing the desired media logic.
  - The Media API provides the capability of creating media pipelines by joining media elements of the toolbox



WebRTC integration



Media pipelines



Part V: Developing for WebRTC

At the client side: PeerConnection API or similar

- No differences with common WebRTC applications
- *… PeerConnection*
- *… Media Stream*
- *… Data Channel*
- Server infrastructure is just another WebRTC end-point.

At the server side: Java EE + Kurento Media API

- Powerful logic available at the application sever
  - *Database integration, JMS, SIP Servlet, REST, SOAP, JCR, etc.*
- App. Logic defines and instantiates the media pipeline in real time
  - *Inspired on JSR 309*
  - *Developer instantiates the desired media elements*
  - *Developer "connects" media elements in the way she likes*
  - *What happens to media formats through the pipeline? Agnostic media*

WebRtcMediaService example

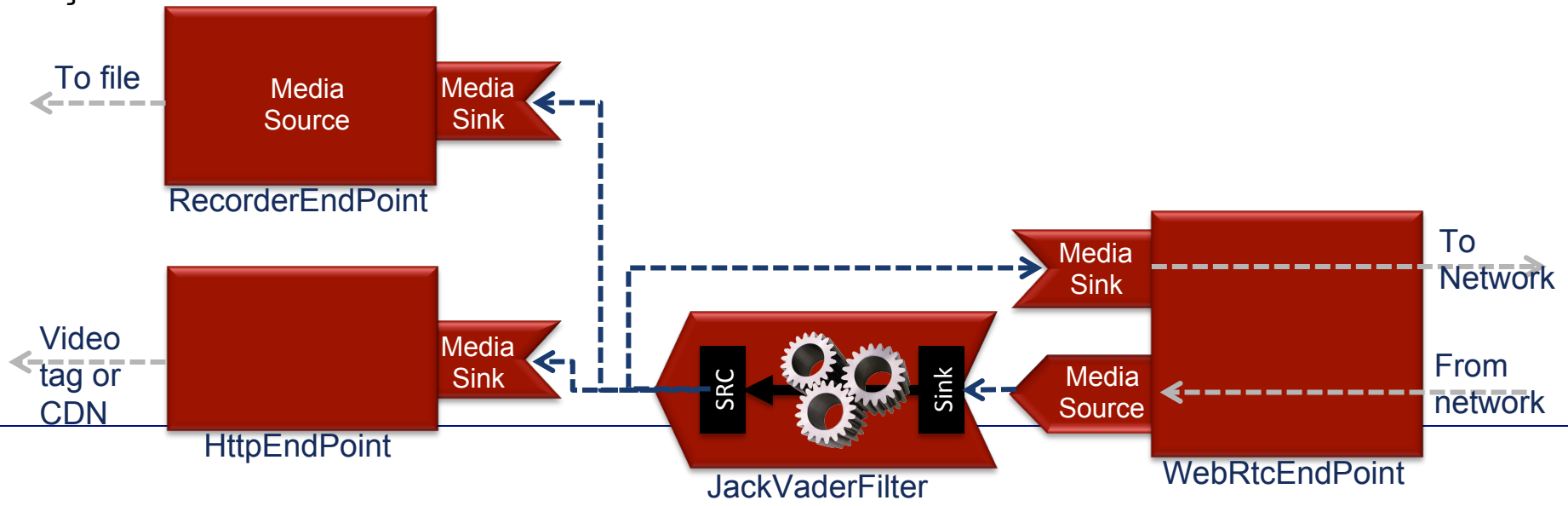
```
@WebRtcMediaService(name = "MyWebRtcService", path = "/pathToService")
public class MyWebRtcService implements WebRtcMediaHandler {

    public void onMediaRequest(WebRtcMediaRequest request) {
        MediaPipelineFactory mpf = request.getMediaPipelineFactory();
        MediaPipeline mp = mpf.createMediaPipeline();

        JackVaderFilter filter = mp.createFilter(JackVaderFilter.class);
        RecorderEndPoint recorder = mp.createRecorder("file:///myFile.webm");
        connect(filter, recorder)

        HttpEndPoint httpEndPoint = mp.createHttpEndPoint();
        connect(filter, httpEndPoint);

        request.startMedia(filter, filter);
    }
```



Media element toolbox

- Transport
  - WebRtcEndPoint
  - RtpEndPoint
  - HttpEndPoint
- Repository
  - PlayerEndPoint
  - RecorderEndPoint
- Group communications
  - MainMixer
  - GridMixer
  - RoundRobinMixer
- Filters
  - FaceRecognitionFilter (events)
  - JackVaderFilter
  - ZBarFilter (QR/Barcode detector with events)
  - PlateRecognitionFilter (events)
  - CDVAMotionDetectorFilter (events)
  - CDVAObjectRecognitionFilter (events)
  - SubtitleFilter
  - ColorTrackingFilter (events)
- And growing …

Part VI: Future of Kurento

Kurento and the Future Internet

- Kurento has been selected by the European Commission for providing multimedia capabilities to FI-WARE: the most important R&D project in Europe in the FI-PPP arena.
- http://www.fi-ware.eu

Kurento shall be available at the FI-LAB

- Cloud infrastructure for FI-WARE open to developers
- More than 800K€ in awards for developers
- More than 100M€ in funding for SMEs start-ups and entrepreneurs
- http://www.fi-ware.eu/lab/

Kurento is and will be FOSS

- LGPL released (https://github.com/kurento)
- www.kurento.org

The End

Luis López

luis.lopez@urjc.es