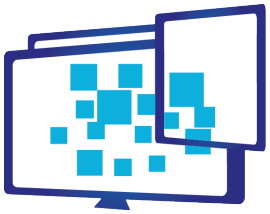
Second Screen User Experiences based upon social networks of people, devices and services

Dave Raggett, W3C

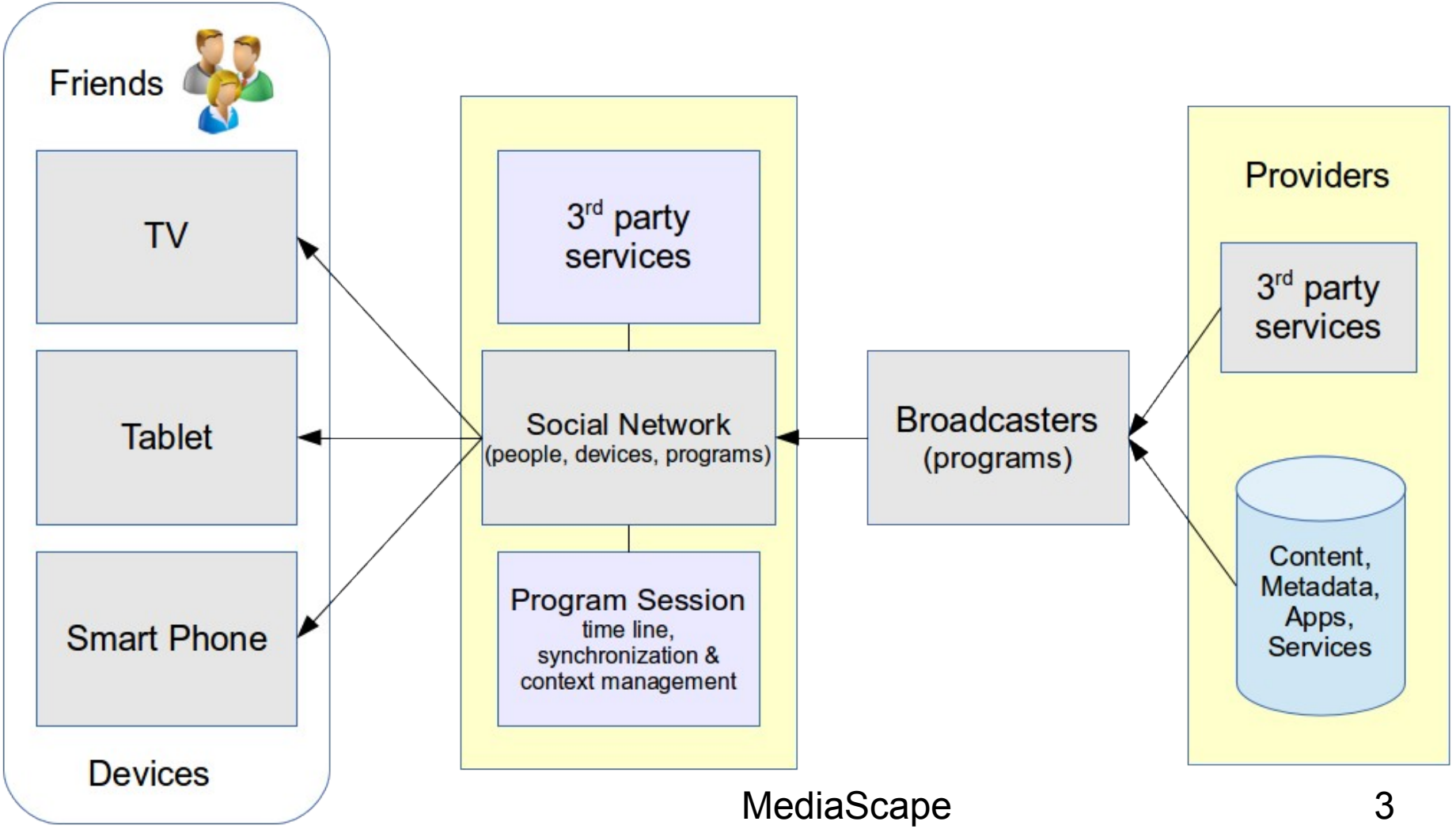
(MediaScape Project)



MediaScape

- European FP7 project
  - Engaging multiscreen single and multi-user experiences for live and timeshifted content
  - Based upon web technologies
- Members include:
  - BBC, Bayerischer Rundfunk, NEC, W3C, Vicomtech-IK4, IRT and Norut
- More at http://www.mediascapeproject.eu/

MediaScape


Friends
TV
Tablet
Smart Phone
Devices
3rd party services
Social Network
(people, devices, programs)
Program Session
time line,
synchronization &
context management
Broadcasters
(programs)
Providers
3rd party services
Content,
Metadata,
Apps,
Services

Use Case*

- John is watching a live sports event on his TV
- John's phone notifies him to start a complementary news service
  - He agrees and it appears as an overlay on his TV
  - He invites his friend Anne to join this service
    - She lives the other side of the city
- She gets the invitation on her phone
  - She accepts
- John and Anne now share the same experience
  - They can exchange comments via their phones

** Many more use cases in the paper!*

Requirements

- Social connection (John and Anne)
- Associating TV and phone
- Allowing phone to know what is being shown on the TV
- Sending notifications across devices
- Adapting user experience to the device
- Synchronizing TV and phone
- Overlaying broadcast content
  - Alternative compositions based upon tiling

How does it work?

- The TV and phone are registered as part of their owner's social network
- Services implemented as HTML5 apps
  - with responsive design for device adaptation
- Social network server tracks context
  - people's status (home, travelling, work, ...)
  - which of their devices are online
  - what programs they are watching
  - Server support for message passing
- Federated social network
  - John and Anne may be on different servers
  - You control your privacy

What's needed?

- Each device needs an agent to be part of the social network
  - HTML Service Workers
    - Execute in background akin to Android services
    - Designed to minimize battery drain on mobile devices
  - Web Sockets (great for async messaging)
- Each app is part of the program session
  - Includes MediaScape JavaScript library
    - Exposes social API and shared context

Synchronization

- Aim: synchronize user experience across devices to within 30 mS using JavaScript
- Social network server measures latency to each device
- Play button sends play request to server
- Server sends play command to devices along with a latency dependent delay
  - So that all devices execute command at same time
- Devices adjust play back rate to their system clock
  - HTML5 audio & video element playbackRate property
- Ask me for a demo in the break

Local vs Remote Messaging

- Local peer to peer discovery and messaging
  - Local network service discovery (DAP WG)
  - Raw Sockets (SysApps WG)
  - NFC handover to WiFi (NFC WG)
  - WebRTC (WebRTC WG)
- But issues with device support and local discovery
- Server based approach is simpler and more reliable
  - Relies on web sockets
  - Avoids challenges of firewalls
  - Users know where devices are in their homes
- JavaScript library can in principle hide differences between local and remote messaging
  - A question of the appropriate abstraction layer

Questions?