WebRTC support in WebKitGTK and WPEWebKit with GStreamer: Current status and plans

Philippe Normand

FOSDEM 2026

About me

- Multimedia engineer at Igalia since ~2009
- Working on GStreamer and WebKit Linux ports
- Was involved in the OpenWebRTC project and its integration in WebKitGTK

igalia

Outline

- Intro
- Current status
- On-going work
- Plans

igalia

Intro - WebKit

- Cross-platform FOSS Web engine, started by Apple as a fork from KHTML
- Powering Safari, GNOME Web, and widely deployed on embedded products (WPE port)
- Multi-process architecture:
  - UIProcess: WebView API consumer
  - WebProcess: DOM, JavaScript, Media (Linux ports), …
  - NetworkProcess: Network handling
  - GPUProcess: WebGL, WebGPU (Apple ports), Media (Apple ports)

igalia

Intro - GstWebRTC

- one GStreamer element (`webrtcbin`) for integration in pipelines
  - GStreamer `RTCPeerConnection` implementation
  - Signals and properties for Offer/Answer negotiation, ICE connection state notifications, etc
- one GStreamer library (`libgstwebrtc.so`)
  - Defines GObject classes for WebRTC-related APIs (Transceivers, …)
- Pluggable ICE and Transport backend support (current default backend uses libnice)

igalia

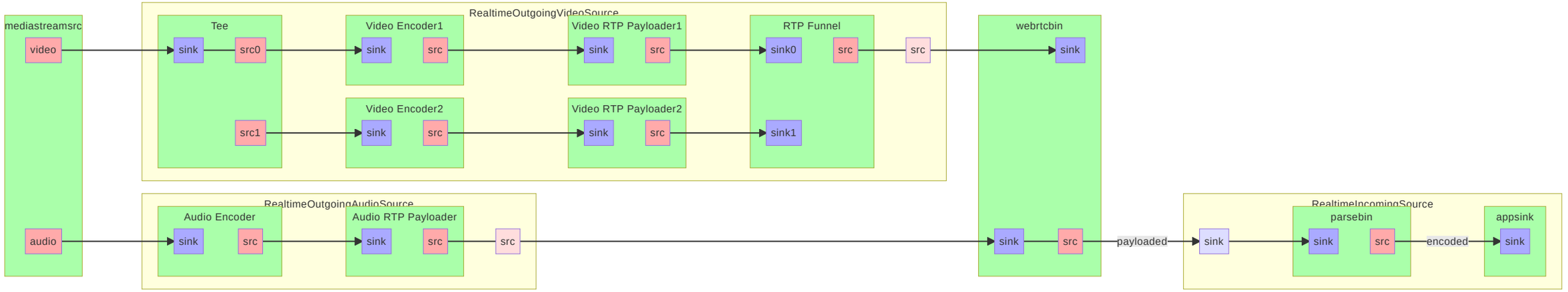
Current status - Capture handling

- One pipeline per capture device (Camera, Microphone, Desktop)
- Permission request handling
- `getUserMedia()` (A/V capture):
  - `Camera` portal support (new in 2025)
  - Fallback: direct access to capture devices using `GstDeviceMonitor`
- `getDisplayMedia()` (Screen capture)
  - Screencasting through the `Desktop` portal

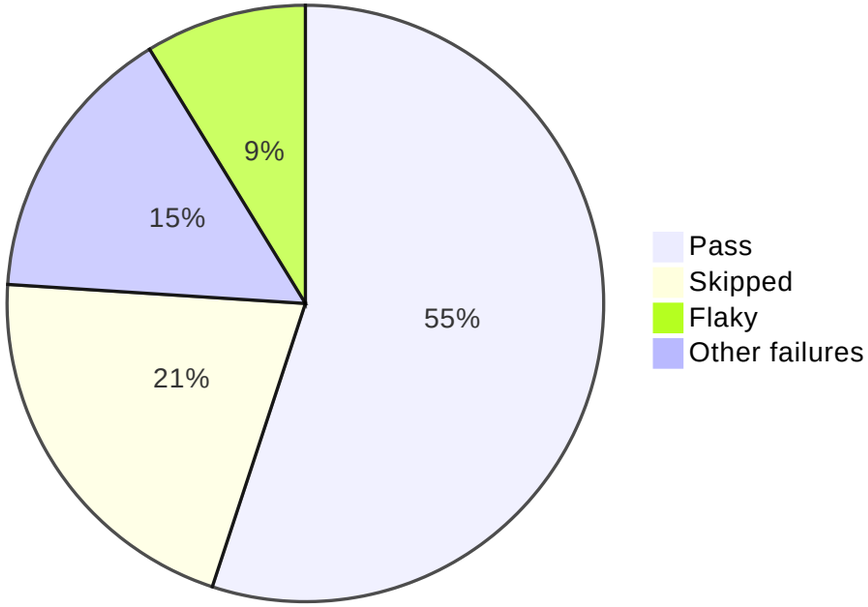
Current status - MediaStream handling

- `webkitmediastreamsrc` GStreamer source element acting as Observer for:
  - Capturer pipelines
  - Canvas / WebAudio MediaStreamAudioSourceNode / `<video>` elements
  - Incoming and Outgoing WebRTC `MediaStreams`
- Each `MediaStreamTrack` maps to a `GstPad` in `webkitmediastreamsrc` and `webrtcbin`

Current status - PeerConnection pipeline



Current status - Tests coverage



Network handling in GstWebRTC: GstWebRTCICE

- GObject base class abstracting access to an ICE agent
- Provides various virtual methods for:
  - STUN / TURN servers configuration
  - ICE candidates handling (gathering, credentials, stats)
  - Transport / Stream handling
- `ice-agent` construct-only property in `webrtcbin`
- Default implementation based on `libnice` used by `webrtcbin`

igalia

Sandboxing the ICE agent, scenarios

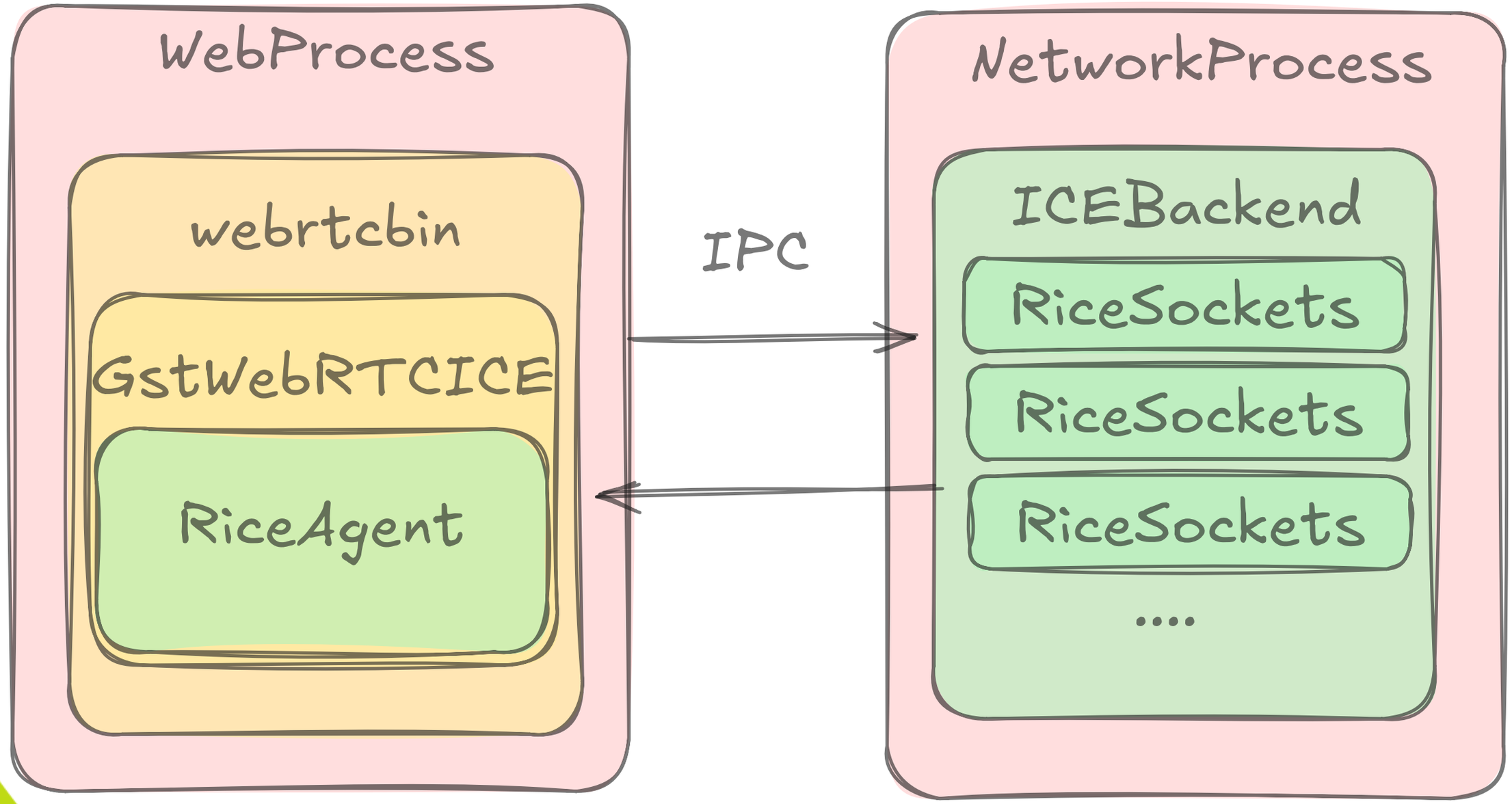
- Sockets abstraction in libnice
- Move the entire libnice agent to NetworkProcess
- LibRice, leveraging its Sans-IO design ;)

LibRice

- Sans-IO implementation of ICE (RFC 8445)
- **https://github.com/ystreet/librice**
- Written in Rust, C API provided
- Two crates:
  - `rice-proto`: Provides the ICE agent API
  - `rice-io` (optional): UDP/TCP sockets handling
- Agent integration in apps using a poll events loop

igalia

Integrating LibRice in WebKit (1/2)


WebProcess
webrtcbin
GstWebRTCICE
RiceAgent
IPC
NetworkProcess
ICEBackend
RiceSockets
RiceSockets
RiceSockets
....

Integrating LibRice in WebKit (2/2)

- Custom GstWebRTCICE agent, with IPC plumbing ✅
- Streaming over UDP sockets ✅
- TURN support 🚧
- TCP sockets handling 🚧
- mDNS support 🚧
  - IETF draft-ietf-mmusic-mdns-ice-candidates-03
  - Ability to resolve FQDNs exposed in 3rd party ICE candidates
  - Expose mDNS FQDN for local IPs in host ICE candidates

igalia

Pending features

- SVC video encoding
- RTP header encryption
- SFrame & Encoded Transforms
- Increased stats coverage
- Transceiver direction changes

igalia

Thanks!