VISA Meeting 2025 @ ALBA Synchrotron

WebX

X11 desktop on the web

Developed as part of the OSCARS European Project

3rd June 2025

Outline

- Motivations
  - Remote Desktop solutions and development ideas
- X11 / Xlib overview
  - Concepts for a Remote Desktop solution
- WebX Engine Design
  - From X11 to ZeroMQ sockets
- WebX System Architecture
  - Remote Desktop in the browser
- WebX Features
  - Fast and efficient image streams with jpeg-turbo (including transparency)
  - Quality levels and window coverage/visibility
  - Messaging
- Demo

Motivations

Not just a nice dev exercise

- Ideas first started in 2019 during PaNOSC
  - Is it possible to record and replay desktop activity (reproducible data analysis)
  - Is it possible to improve on Guacamole:
    - Why encode/decode into base64 rather than use binary data?
    - Why does moving windows around produce so much data/use so much CPU?

- Development started as a PoC but quickly showed promise
  - CPU usage seen to be low
  - Latency (compared to Guacamole) very good
  - User experience much better
    - Window *tearing* gone
    - Image quality good
    - Interactions more fluid

- Development over the years has been sporadic
  - Session management
  - Multi-client and variable bandwidth managed
  - Now ready for production use

Motivations

Alternative Open-Source Remote Desktop Solutions

- Analysis of existing Open Source Remote Desktop solutions
  - Guacamole
  - Rustdesk
  - NoVNC
  - Selkies-Gstreamer
  - KasmVNC
- Often good quality products with nice features
  - Eg KasmVNC has resizing and multi-screen support
- None provide a perfect solution for integration into VISA
  - Some missing web clients
  - Some missing relay capabilities (needed to manage access rules/notifications)
  - Some missing session management and multi-client support
  - All (except NoVNC) use 100% CPU when moving windows or during graphical updates

X11 / Xlib overview

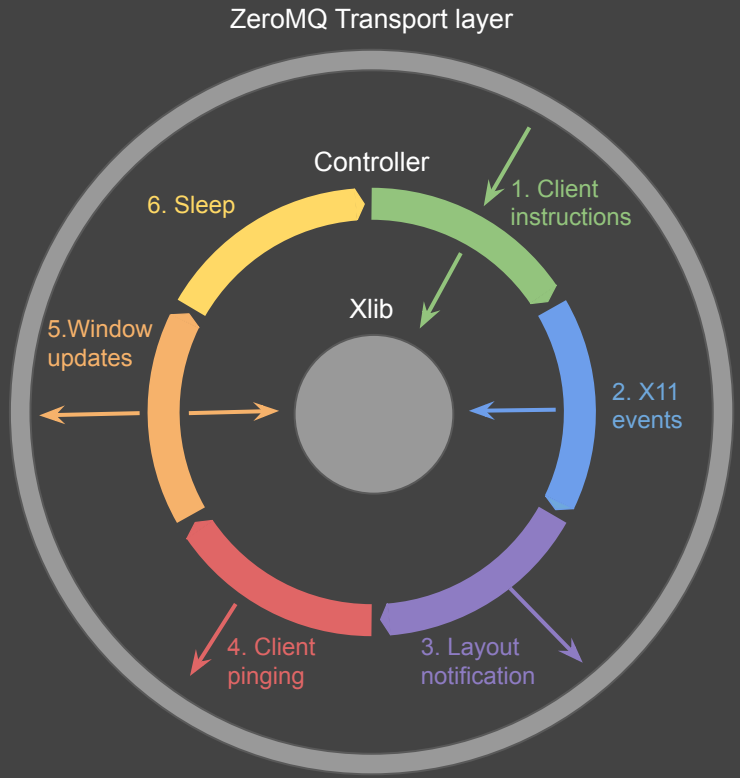
Growing ideas for WebX

- 2 main concepts behind WebX:
  - Obtain a list of windows including their position and sizes
  - Obtain graphical content of each window
- Xlib provides low-level access to X11 window management to:
  - Obtain a tree of windows
  - Obtain image data for window (full or partial)
  - Subscribe to change notifications
    - Window creation/deletion
    - Window move/resize
    - Window *damage* (content modification)
  - Handle mouse events
  - Handle keyboard events
- Provides simple basis to develop the WebX Engine: a window-based remote desktop solution
  - Obtain window information from Xlib
  - Transfer information to connected clients
  - Avoid the common effect seen in other solutions where moving windows around a screen consumes CPU and high bandwidth

WebX Engine Design

From X11 to ZMQ

- ZeroMQ concurrency framework
  - Network patterns over sockets
  - publisher/subscriber & request/response
- WebX Engine control loop:
  - Client instructions
    - mouse and keyboard
  - X11 window events
    - Created / Deleted
    - Movement / Resize
    - Damage
  - Notify clients of layout changes
  - Client pinging and quality check
  - Client window updates
    - Send image data (timed)
  - Sleep
- Runs at 60fps
  - Client window updates less frequent



WebX Engine Design

Client management

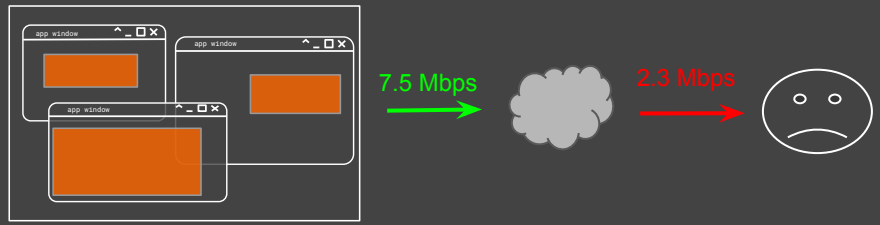
- Clients grouped by *Quality level*
  - Window content refresh rate
  - Image quality
  - Bitrate capping (per window)
- Each client-group maintains a representation of the desktop
  - All visible windows containing:
    - *Damaged* area
    - Quality handler (bitrate dependent)
  - Calculate total Mbps for all window updates
- Need to determine which group a client belongs to

Same window layout, different *damage* areas, data rates and quality

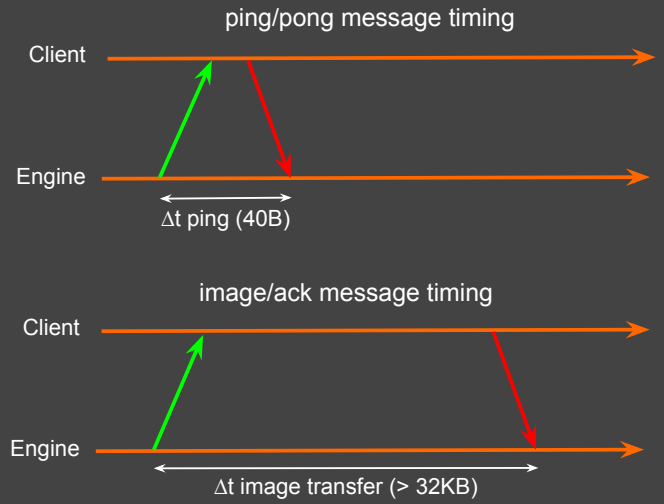
WebX Engine Design

Client management

- Client pinging
  - Ensures client list is always valid
  - Determines client latency
- Estimation of client bitrate
  - Measure timing of image transfer
  - Automatic adjustment to client group

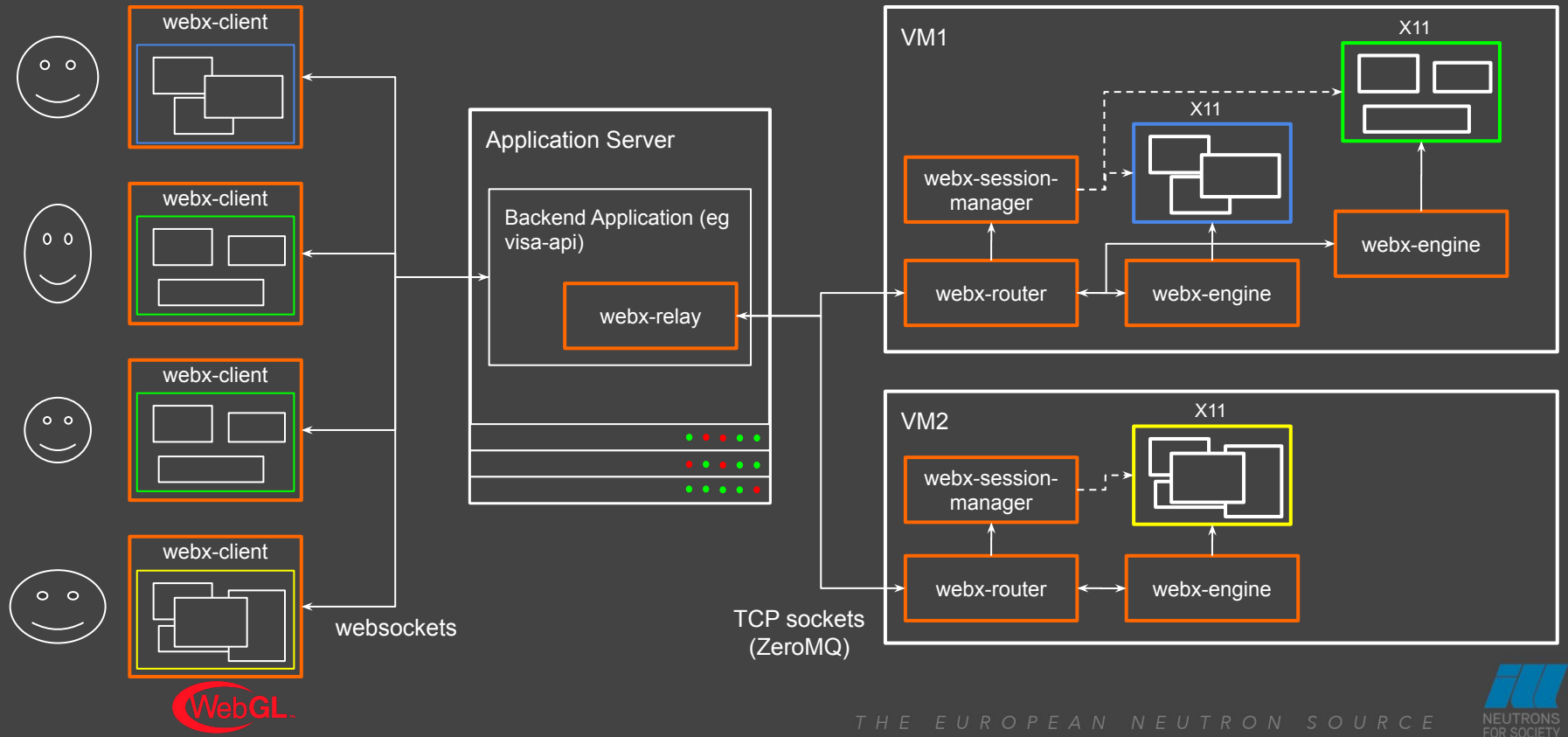


- Downgrade client to lower quality if group egress bitrate close to or above client ingress bitrate



WebX System Architecture

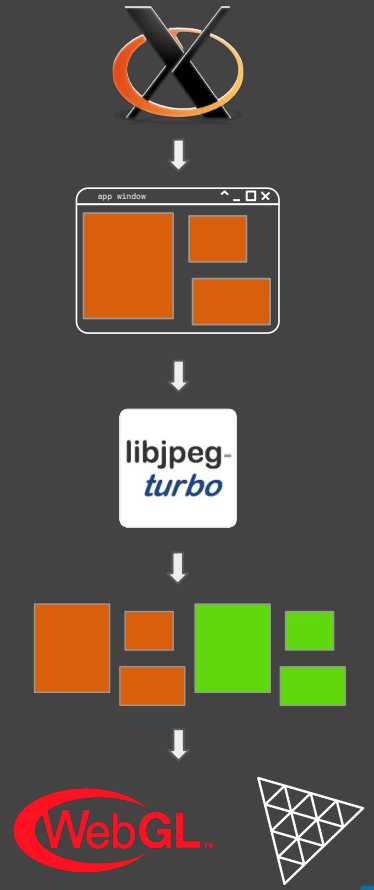
How the Remote Desktop appears in a browser



WebX Features

Fast and efficient image streaming with libjpeg-turbo

- X11 produces window damage events
  - Graphical updates to parts of windows (rectangles within the window)
- WebX combines the rectangles over *N* controller loops
  - Clients notification time depends on quality level
- Xlib grabs raw window rectangle bitmaps
- Libjpeg-turbo converts/compresses images
  - Uses SIMD instructions for extremely quick conversion (~4ms)
  - Really low CPU usage
- But Jpeg has no alpha channel… how to handle transparency?
  - Hack the initial RGBA bitmaps to remove RGB
  - Offset pointer to image data by 2 bytes (shift alpha channel to green)
  - Generate a second Jpeg: an *alpha-map* used by *three.js* in the webx-client
- Send combined images in single binary message to clients
- Recombine images using WebGL



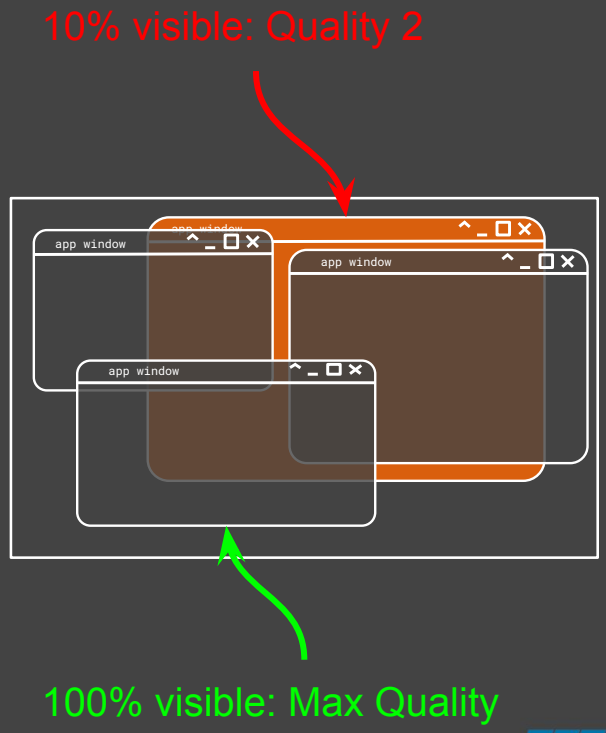
WebX Features

Quality levels and window coverage/visibility

- 12 pre-defined quality levels

	Highest Quality	Lowest Quality
Window refresh fps	30	0.5
Image quality	0.9	0.4
Data rate	24 Mbps	0.5 Mbps

- Each window measures generated bitrate
  - Reduces quality if data rate exceeded
- Window coverage reduces quality
  - Mouse over window increases quality

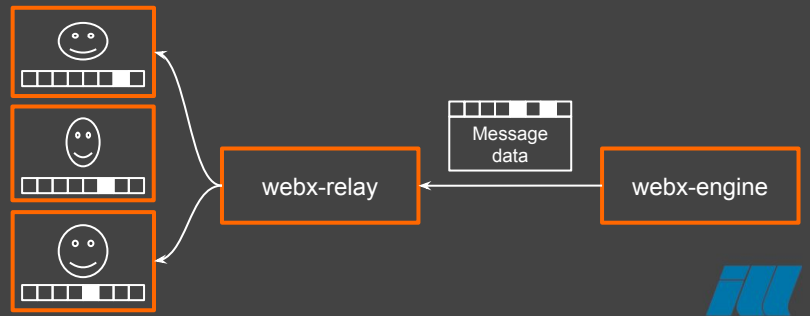


WebX Features

Messaging

- All messages are binary
  - Efficient and very quick serialisation and deserialisation
- Each client has a unique Index
  - 8 bytes => 64 clients
- Each message contains
  - bitmask corresponding to destination clients
  - Unique Session Id (16 bytes)
- Webx-relay filters clients that match bitmask and Session Id
  - Messages can be directed to individuals (pings), groups (window images) or all (window layout)

	Content	Usage
Screen message	Screen size, Max Quality	Initialises the client
Image and SubImage Messages	Full window image or partial window images	Sent when window content changes
Windows message	Window layout (coordinates and sizes)	Sent of window move / resize
Mouse and Cursor Image Messages	Mouse position / cursor Id and Cursor Jpeg	Sent when mouse moves or cursor changes
Ping	Timestamp	Ensure client connected and determine latency
Clipboard	X11 Clipboard content	Synchronisation with client clipboard



VISA Meeting 2025 @ ALBA

Thanks

Time for a demo