

# Constraints (constrainable properties) are

- A cross between
  - What browser implementers need/can offer
  - What application developers need

# The HTML logical document model

- `<html> hello world </html>` is valid
- Only because the browser does smart things
- Web success is due in part to browser default intelligence

# What implementers need/can offer

- Implementer needs
  - For gUM, must work with shared devices
    - E.g., shared camera across apps, browsers, tabs
  - For WebRTC, unpredictable networks
    - Transport or negotiation may fail
- With flexibility, implementers can offer
  - Improved browser performance
    - E.g., selection of native camera resolution to simplify scaling
  - Improved network performance
    - e.g., change in codec or even native video resolution, size, or frame rate can help congestion

# What application developers need

- Predictability/controllability to the level that matters for their app
- For critical aspects, alerting when the browser cannot provide it

# The constraint approach

- Each capability has a set or range of allowed values
- Developer indicates
  - Which subset of values or ranges is acceptable
  - Whether an error event must be raised if the constraint cannot be satisfied ("mandatory")
  - For non-critical ("optional") constraints, which ones are more important by priority ordering
- Browser then
  - Is free to select or change a setting for each capability as long as it satisfies the constraint
  - Throws an exception for any inability to satisfy a mandatory constraint

# Sample use cases

- Broadcasting US Presidential address
  - Camera aspect ratio must not change (mandatory)
  - Prefer highest resolution but will accept lower
  - Any refresh rate okay
- Door security camera
  - Want best of resolution, refresh rate, etc. (so no mandatory constraints)
  - Prefer high refresh rate to high resolution if one must be sacrificed

# Constraints vs. direct configuration

- Use constraints when
  - The browser does not have direct and exclusive control over the capability, or
  - The browser can provide some benefit if allowed flexibility to choose among developer-acceptable possibilities
- Use direct configuration when
  - The browser has direct and exclusive control over that capability, and
  - The browser cannot add benefit through being allowed to choose from among a set of alternatives

# Types of constraints (so far)

- Property value range (min/max)
  - Video width, height, aspect ratio, etc.
- Enumerated list
  - Source id, <boolean constraints>



# Constraints, capabilities, and settings

- Capabilities:
  - Maximum ranges or possible values for a constrainable property
- Constraints:
  - Restricted ranges or possible values for a constrainable property,
  - Indication of mandatory or optional, and
  - If optional, priority relative to other constraints
- Settings:
  - The actual current values for constrainable properties
- Jim will cover in more detail