Parent or "Chrome" Process   Child or "Content" Process
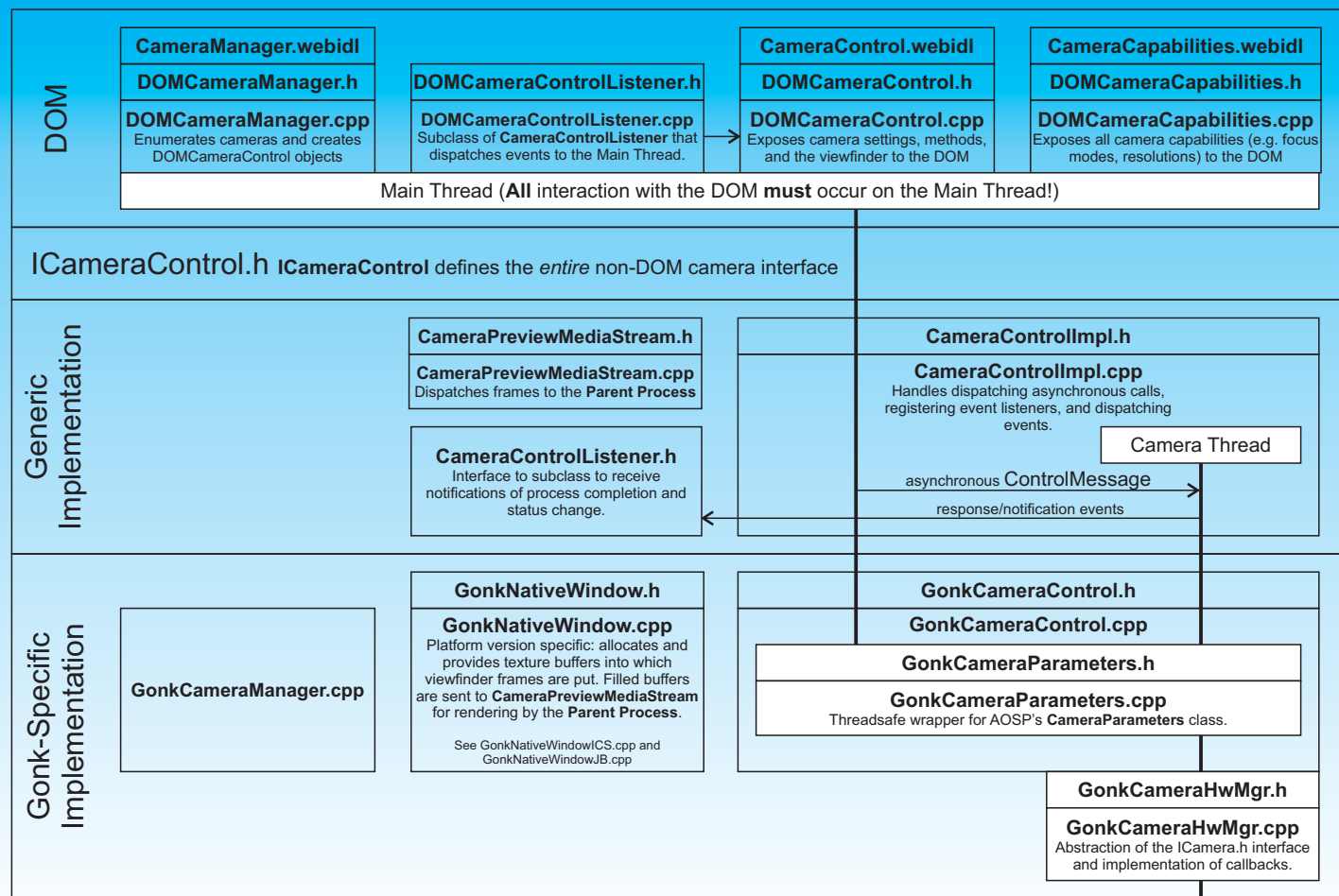
Gaia: HTML, JS, CSS

| System and Window Management | Camera application |

Gecko: C++

Parent or "Chrome" Process

DOM

CameraManager.webidl		CameraControl.webidl	CameraCapabilities.webidl
DOMCameraManager.h	DOMCameraControlListener.h	DOMCameraControl.h	DOMCameraCapabilities.h
DOMCameraManager.cpp Enumerates cameras and creates DOMCameraControl objects	DOMCameraControlListener.cpp Subclass of **CameraControlListener** that dispatches events to the Main Thread.	DOMCameraControl.cpp Exposes camera settings, methods, and the viewfinder to the DOM	DOMCameraCapabilities.cpp Exposes all camera capabilities (e.g. focus modes, resolutions) to the DOM

Main Thread (**All** interaction with the DOM **must** occur on the Main Thread!)

ICameraControl.h **ICameraControl** defines the *entire* non-DOM camera interface

Generic Implementation

| CameraPreviewMediaStream.h | CameraControlImpl.h |
| CameraPreviewMediaStream.cpp
Dispatches frames to the **Parent Process** | CameraControlImpl.cpp
Handles dispatching asynchronous calls, registering event listeners, and dispatching events. |

Camera Thread

CameraControlListener.h
Interface to subclass to receive notifications of process completion and status change.

asynchronous ControlMessage

response/notification events

Gonk-Specific Implementation

| GonkNativeWindow.h | GonkCameraControl.h |
| GonkNativeWindow.cpp
Platform version specific: allocates and provides texture buffers into which viewfinder frames are put. Filled buffers are sent to **CameraPreviewMediaStream** for rendering by the **Parent Process**.

See GonkNativeWindowICS.cpp and GonkNativeWindowJB.cpp | GonkCameraControl.cpp |

GonkCameraManager.cpp

GonkCameraParameters.h

GonkCameraParameters.cpp
Threadsafe wrapper for AOSP's **CameraParameters** class.

GonkCameraHwMgr.h

GonkCameraHwMgr.cpp
Abstraction of the ICamera.h interface and implementation of callbacks.

Gonk: AOSP, Linux, etc.

ICamera.h
Calls made to the functions defined in ICamera.h are sent over an IPC channel to the **mediaserver** process, which runs the vendor-specific implementation of **CameraHardwareInterface**.