Google™ 09 I/O

Noisy Androids

Mastering the Android Media Framework

Dave Sparks
27-May-2009

Agenda

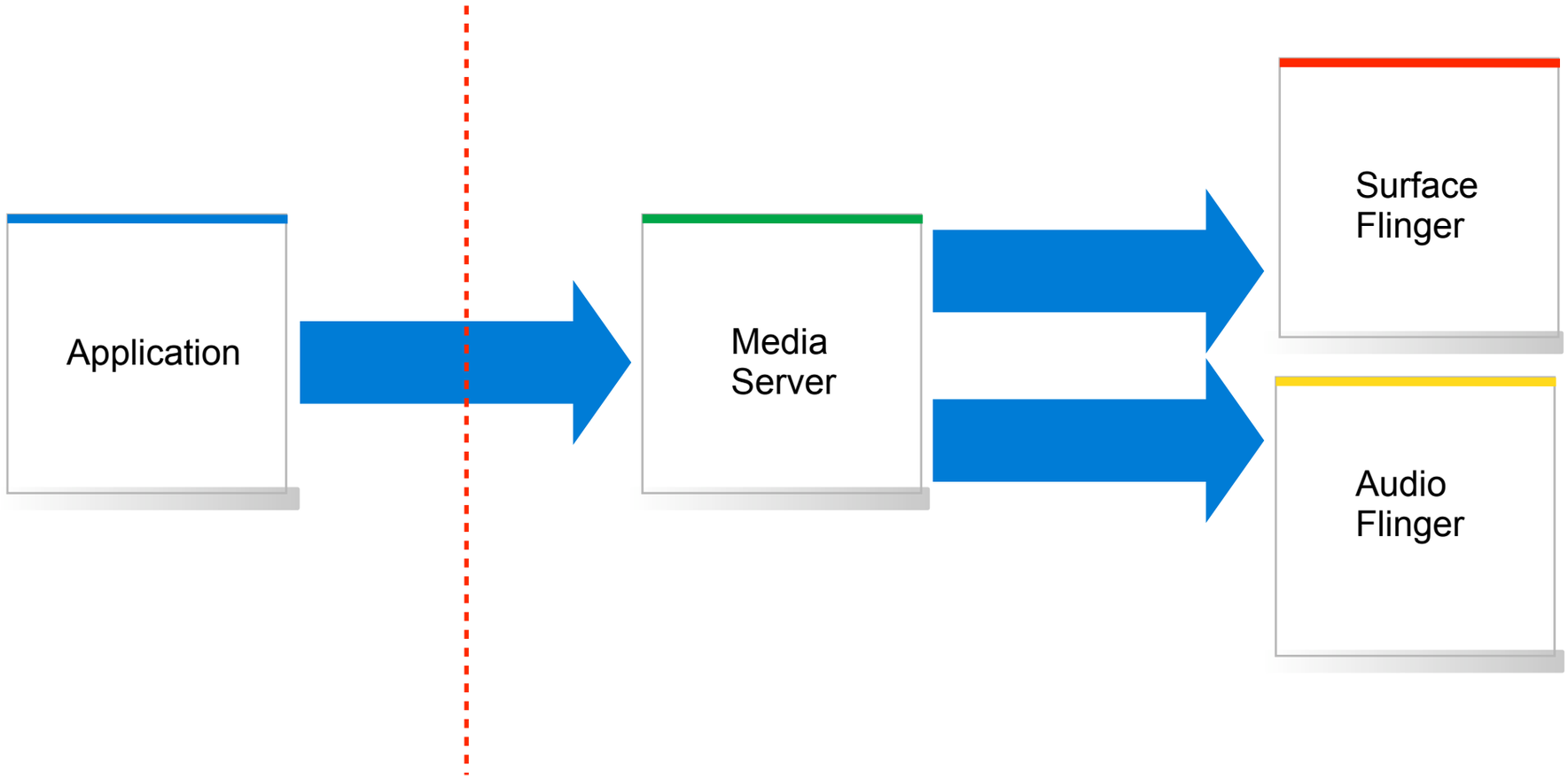
- Frank Lloyd Android: Media Framework architecture
- Sweet Android: What's new in Cupcake V1.5?
- Busted Android: Common problems
- Curious Android: Q & A

Architecture

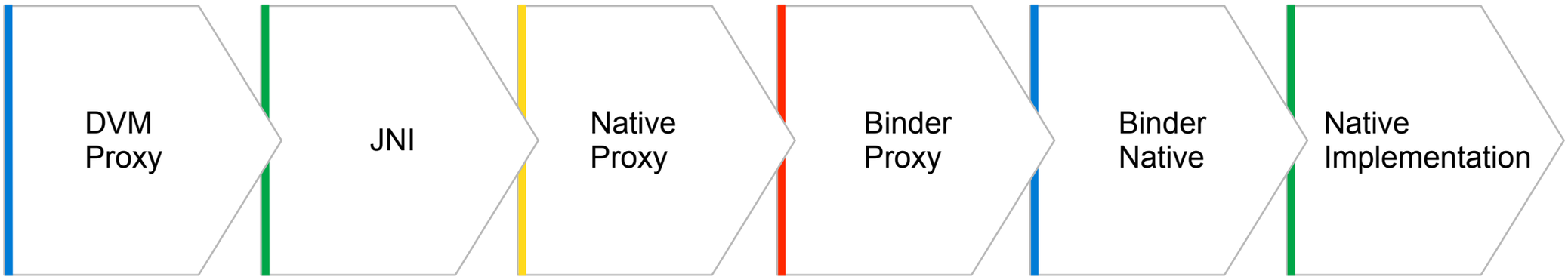
Design Goals

- Simplify application development
- Share resources in multi-tasked environment
- Strong security model
- Room for future growth

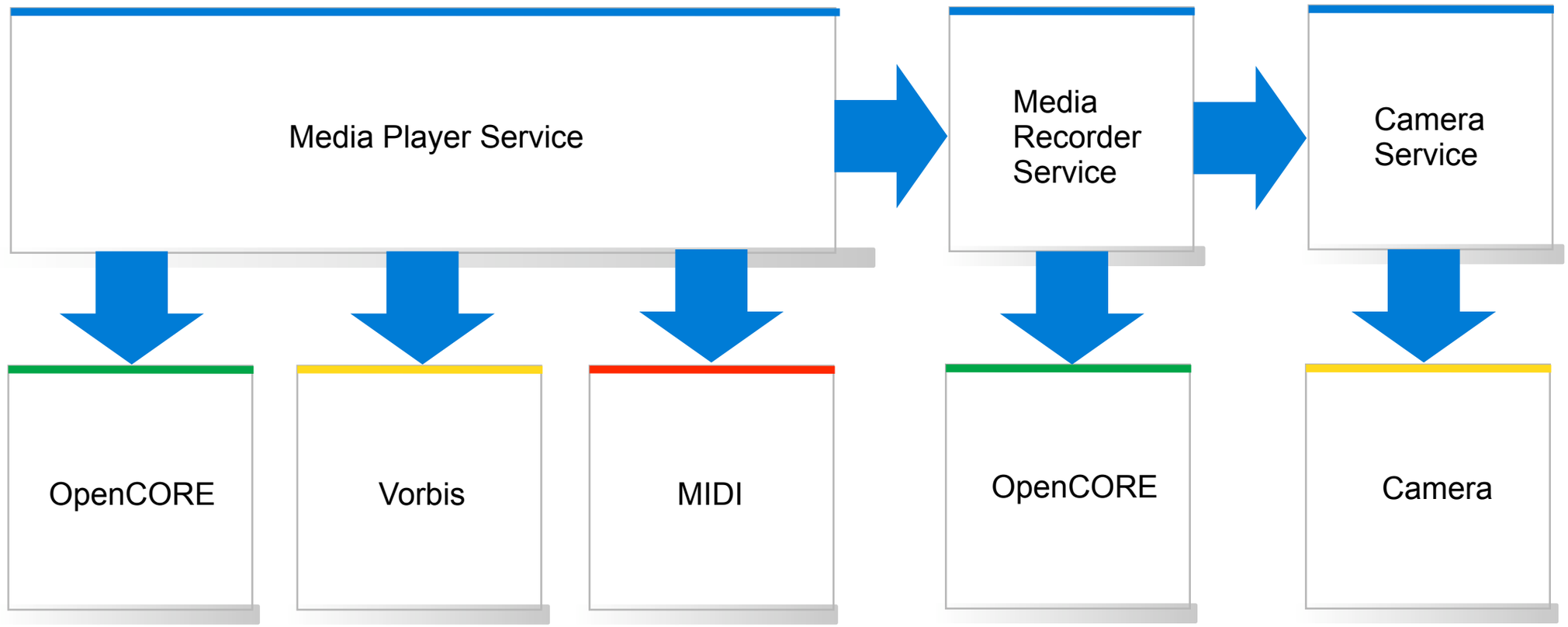
Media Framework



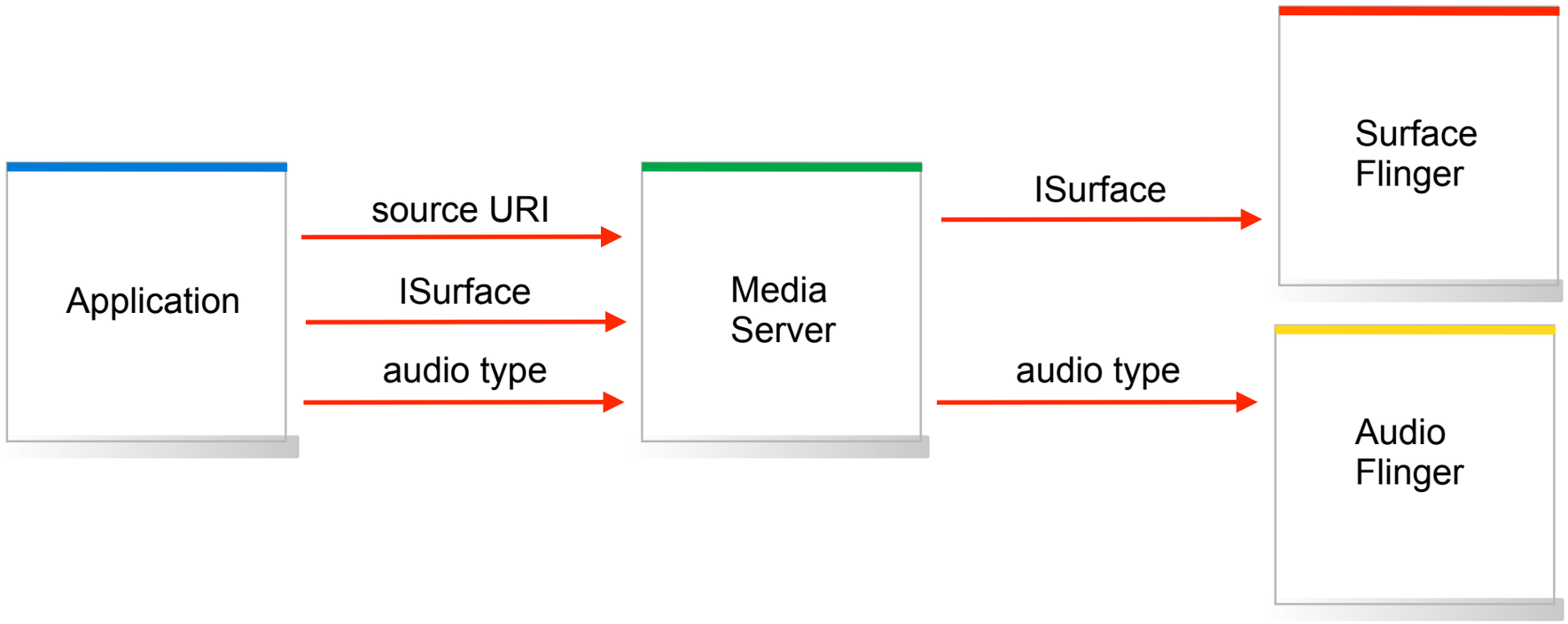
Typical Stack for Media Function Call



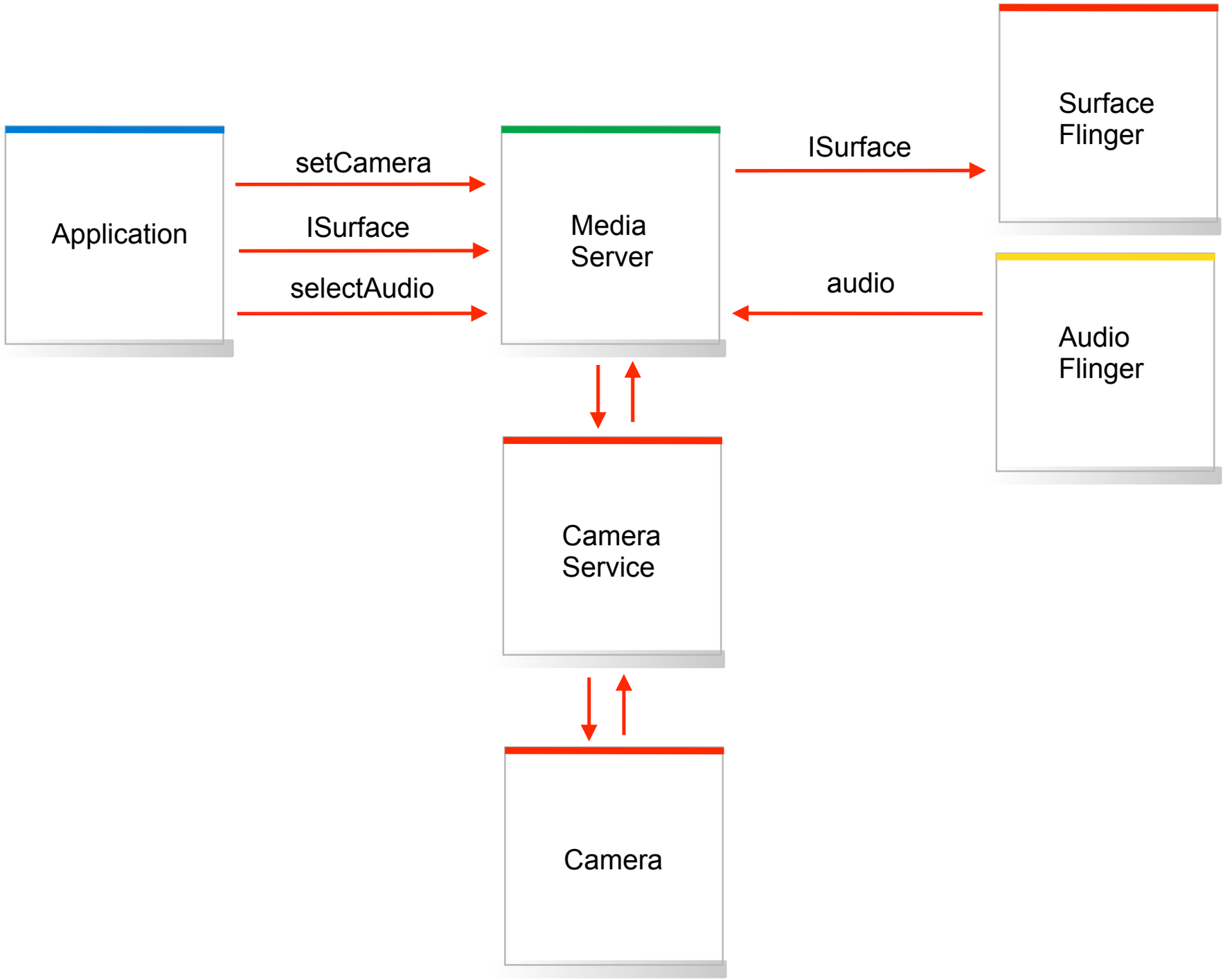
Media Server Process



Media Playback



Media Recorder



Codecs

H.263 Video

- Originally designed for low bit-rate video conferencing
- Simple encoder
- 3GPP standard
- Used by many streaming sites for low bit-rate video

MPEG4-SP (Simple Profile) Video

- Designed as replacement for MPEG1/2 codecs
- Simple encoder
- Is it really an improvement over H.263?

H.264 AVC (Advanced Video Codec)

- Better compression (e.g. multiple reference frames)
- Better quality (e.g. mandatory in-loop deblocking filter)
- Different profiles target different applications and devices
- Uses include digital cinema, HDTV broadcast, and mobile
- More complex than H.263 or MPEG4-SP

MP3

- First generation psycho-acoustic compression
- Approximately 10:1 compression @ 128kbps
- Sonic transparency 192Kbps

AAC (Advanced Audio Codec)

- Psycho-acoustic compression like MP3
- Better compression than MP3
- Sonic transparency 128Kbps
- Commonly used in MPEG-4 streams

Ogg Vorbis

- Psycho-acoustic compression like MP3
- Better compression than MP3
- Low-overhead player
  - Lower latency
  - Uses less memory
- Can loop seamlessly (unlike MP3)

Adaptive Multi-rate (AMR) Audio

- Two flavors: Narrow band and wide band
- Narrow band is 8KHz input, bit-rates 4.75K to 12.2K bps
- Wide band to 16KHz input, bit-rates 6.6K to 23.85K bps
- Used in 3GP streams

Streams

Typical 3GPP Stream

- Lower quality
- H.263 video codec
- AMR-NB audio codec
- Bit rates up to 192K bps

Typical MPEG-4 Stream

- Usually higher quality than 3GPP
- H.264 video codec
- AAC audio codec
- Bit rates up to 500K bps

What container/codec type should I use?

- Authoring for Android device, use MPEG4 container with H.264 video (HVGA up to 500Kbps) and AAC 96Kbps
- Creating content on Android device for other Android devices: Use 3GPP container with H.263 video (CIF up to 384Kbps) and AMR-NB audio
- Creating content on Android device for other devices: Use 3GPP container with H.263 video (QCIF up to 192Kbps) and AMR-NB audio
- Make sure that ‘moov’ atom is located before ‘mdat’ atom for HTTP progressive streaming

New Features in V1.5

New Media Features For Cupcake V1.5

- Video recording
- AudioTrack
- AudioRecord
- JET interactive MIDI engine

AudioTrack and AudioRecord

- Expose raw PCM audio streams to applications
- AudioTrack: Write PCM audio directly to mixer engine
- AudioRecord: Read PCM audio directly from mic
- Callback mechanism for threaded applications
- Static buffer for playing sound effects

JET Interactive MIDI Engine

- Based on MIDI - file sizes can be small
- DLS support allows for better quality instruments
- Precise synchronization for layered MIDI tracks
- Native code - very efficient
- Synchronization callbacks to applications for rhythm games
- Open source engine and creation tools
- VST plugin - use it inside your favorite DAW tool

Common Problems

Problem: Volume control behavior is inconsistent

- Volume control is overloaded
- AudioManager has a default strategy for controlling volume:
  - If in-call, adjust in-call volume
  - If ringing, mute ringer
  - If media track is active, adjust media volume
  - Otherwise, adjust ringtone volume
- In an application that plays sounds periodically, the volume behavior is not consistent

Solution: Set the default stream type

```
import android.app.Activity;
import android.media.AudioManager;
...
public void onCreate(Bundle icicle)
{
  super.onCreate(icicle);
  setVolumeControlStream(AudioManager.STREAM_MUSIC);
  ...
}
```

Problem: Unable to play file from resource?

```
MediaPlayer mp = new MediaPlayer();
try {
  mp.setDataSource(“res:com.myapp.raw.test”);
  mp.prepare();
} catch (IOException e) {
  Log.e(“Error ” + e.print() + “ opening media player”);
}
```

Solution: Use AssetFileDescriptor

```
AssetFileDescriptor afd =
  context.getResources().openRawResourceFd(resId);

set.DataSource(afd.getFileDescriptor(),
               afd.getStartOffset(),
               afd.getLength());

afd.close();
```

Problem: Out of MediaPlayers!

```
MediaPlayer mpArray = new Object[50];
for (int i = 0; i < 50; i++) {
  MediaPlayer mp = MediaPlayer.create(soundName[i]);
  mp.prepare();
  mpArray[i] = mp;
}
```

Solution: Reuse MediaPlayer

- Call release() and set to null
- Or call reset(), then setDataSource()
- Limit to 2 or 3 maximum

Problem: CPU Overloaded

```
MediaPlayer sound1 = new MediaPlayer();
MediaPlayer sound2 = new MediaPlayer();
MediaPlayer sound3 = new MediaPlayer();
// call setDataSource, prepare, etc.
...
// later on...
sound1.start();
sound2.start();
sound3.start();
...
// CPU is bogging down here...
```

Solution: Use SoundPool

```
import android.media.SoundPool;
import android.media.AudioSystem;

Context context = getApplicationContext();
sp = new SoundPool(maxStreams, AudioSystem.STREAM_MUSIC, 0);
int snd1 = sp.load(context, res.Raw.pow);
int snd2 = sp.load(context, res.Raw.blam);
int snd2 = sp.load(context, res.Raw.biff);
...
sp.play(snd1, leftVol, rightVol, priority, loop, rate);
...
sp.play(snd2, leftVol, rightVol, priority, loop, rate);
...
sp.play(snd3, leftVol, rightVol, priority, loop, rate);
```

Q & A

Google™ 09 I/O

Security

Media Server Security

- Isolate file parsers and codecs in a separate process
- Centralize hardware resource management and security
- Enforce permissions on camera and audio record
- Media server runs at reduced privileges
- Applications grant file access on as-needed basis

Media Server Security (continued)

- Media server has read/write access to SD card only
- Applications typically pass in an open file descriptor
- Only the media server process has direct access to hardware resources
- Applications must include permissions in their manifest to access camera and microphone

How does sandboxing defeat a hypothetical attack?

- Assume we have a buffer overrun vulnerability in a codec
- Hacker creates a malicious stream that exploits vulnerability
- Assume that hacker can run arbitrary code in the media server process
- Media server can access SD card, but not data or system partition
- Exposed data from data partition is limited to files that applications have explicitly opened for the media server
- DOS attacks are possible, but limited in scope due to sandboxing
- No data from application is vulnerable

Google™ 09 I/O

Why Do I Need to Use This Template?

- We all want to present the same picture of Google
  - As a unified company with a single brand promise
- By standardizing on a common template, slides can be shared and moved between files with no disparity in their look and feel
- By standardizing on a common template, slides can be shared and moved between files with no disparity in their look and feel

Presentation Bullet Slide Layout

- Use this template to create your presentation
- Use the provided color palette, box and arrow graphics, and chart styles
- Instructions are provided to assist you in using this presentation template effectively
- At all times, strive to maintain Google’s corporate look and feel

Bullet Slide With Subtitle Placeholder

Chart title or subtitle placeholder

- Select the subtitle text box above, copy, and paste it on to the slide that requires a subtitle
  - Vertical position setting for subtitle is 1.01"
  - Vertical position setting for bullet text is 1.5"
  - This way the subtitles will be in the exact position throughout the presentation and will not jump from slide to slide
- Subtitle capitalization is sentence case
- Subtitles never have a period at the end

Code Slide With Subtitle Placeholder

Chart title or subtitle placeholder

```
var map = new GMap2(document.getElementById("map"));
map.setCenter(new GLatLng(37.442, -122.142), 13);
var bounds = map.getBounds();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i = 0; i < 10; i++) {
  var point = new GLatLng(southWest.lat() + latSpan *
    Math.random(), southWest.lng() + lngSpan * Math.random());
  map.addOverlay(new GMarker(point));
}
```

Color Palette

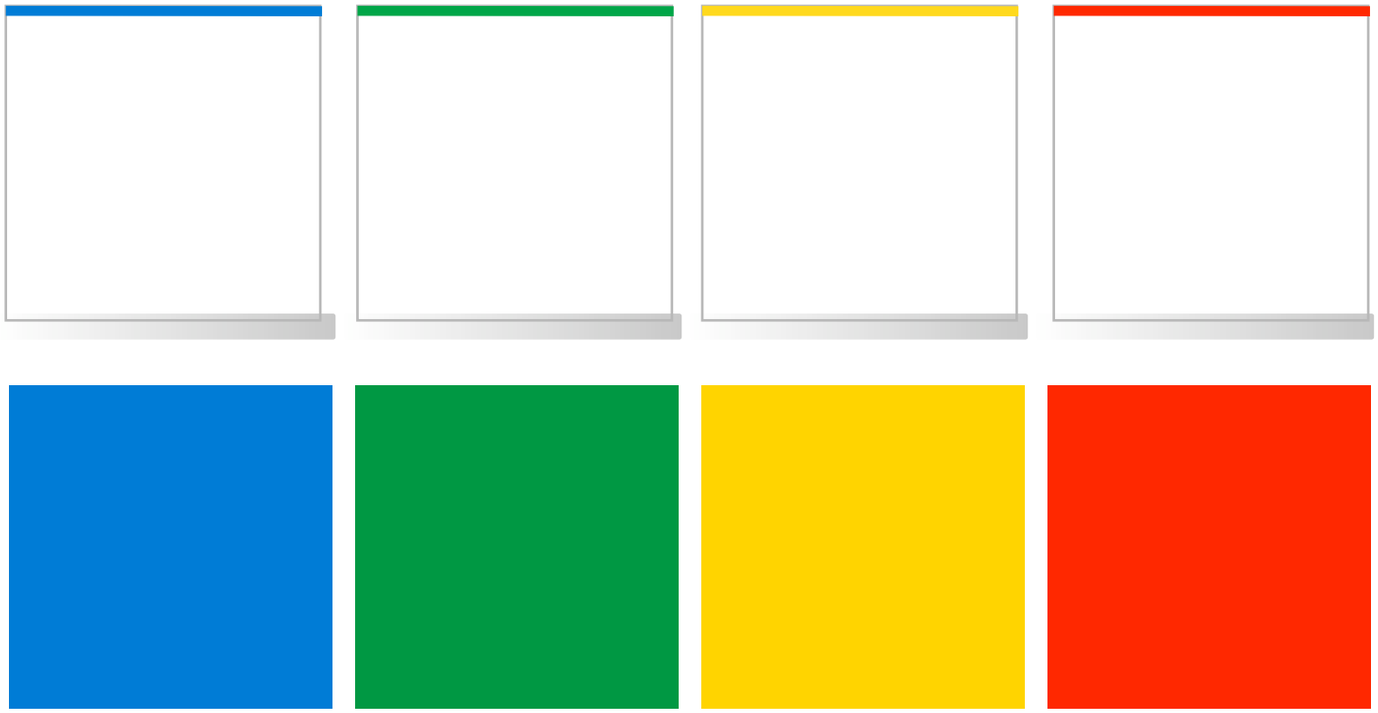
Primary

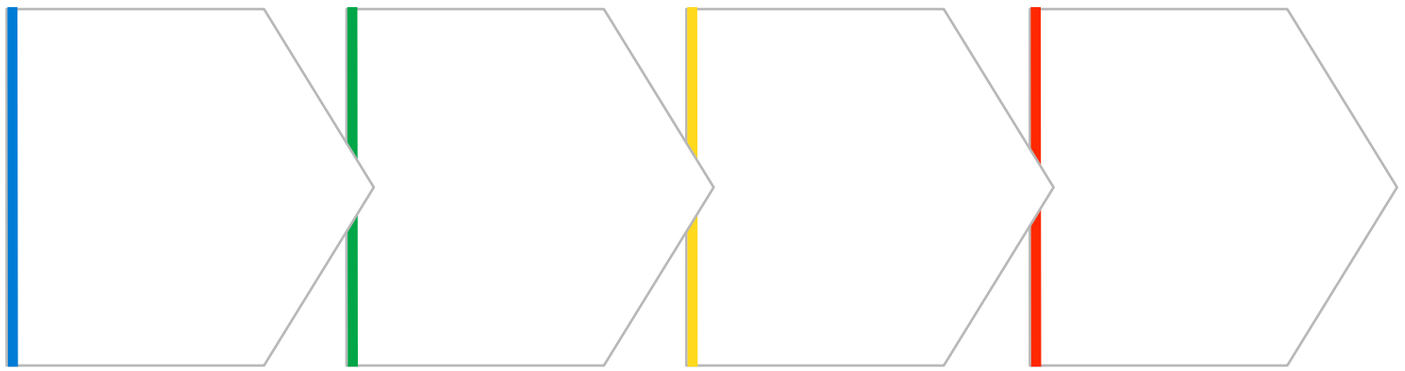
Neutral

Picker

Google 09 I/O

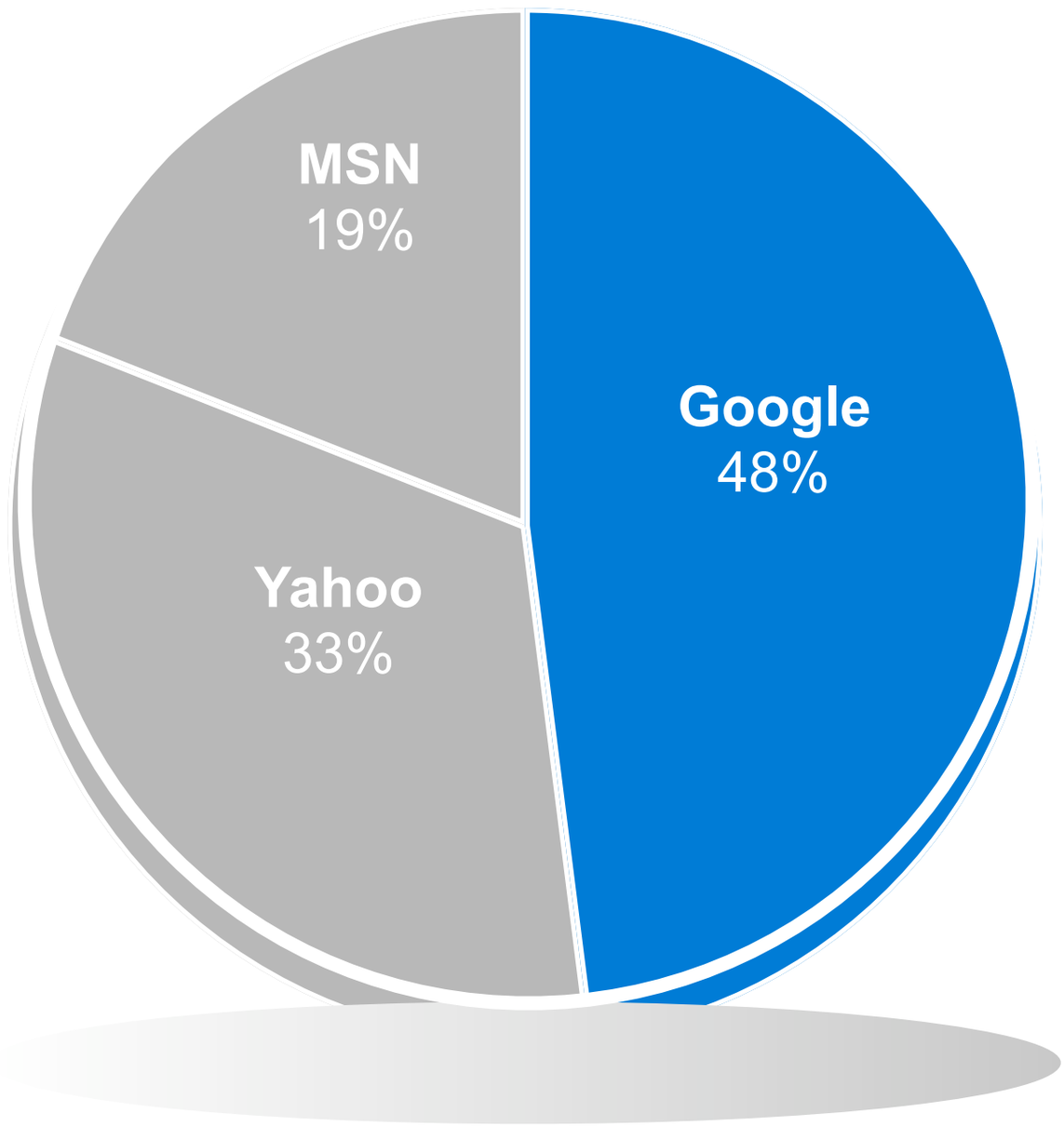
Graphic Element Styles and Arrows





Pie Chart Example

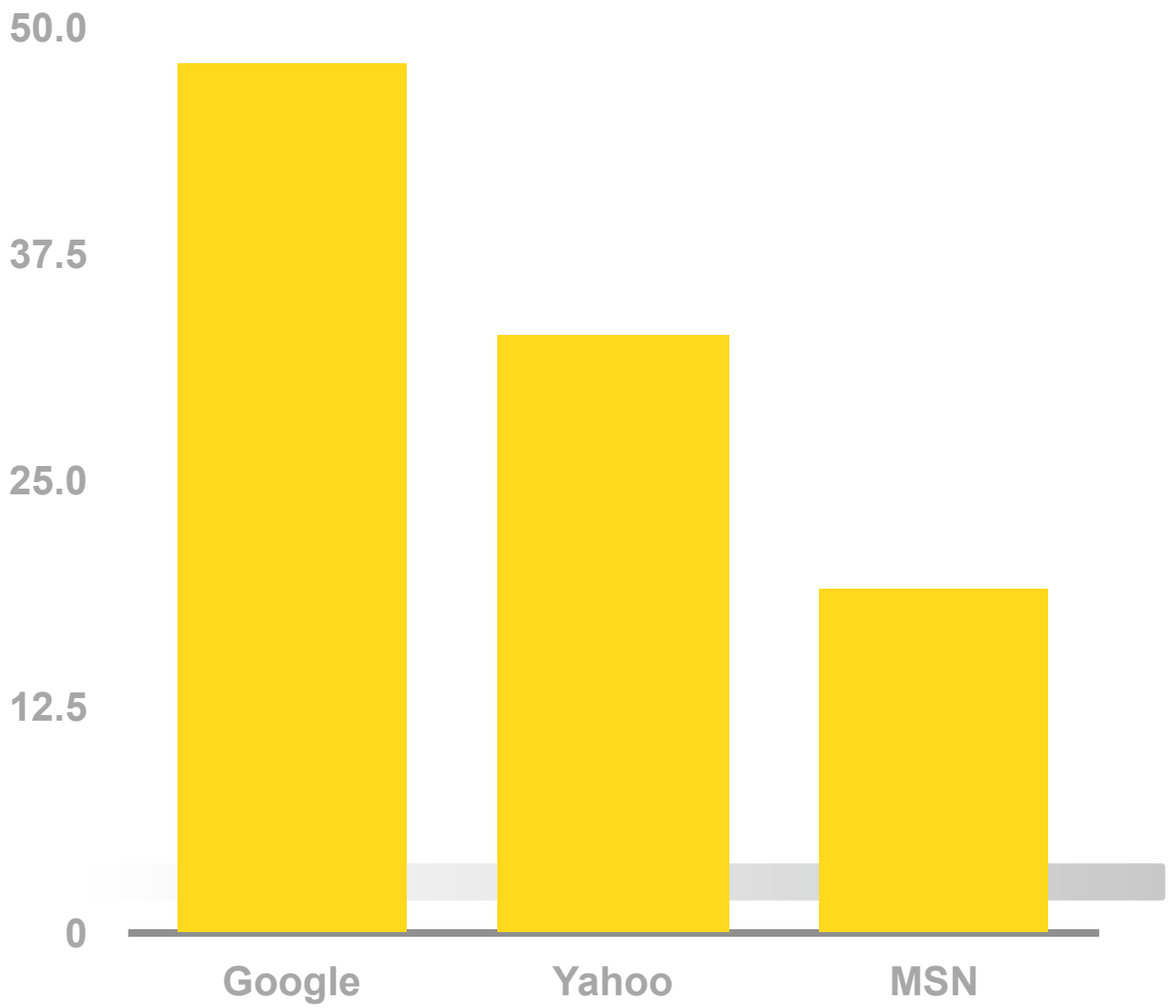
Chart title or subtitle placeholder



Source: Insight Express Survey of 1000 home Internet users (June 2004)

Column Chart Example

Chart title or subtitle placeholder

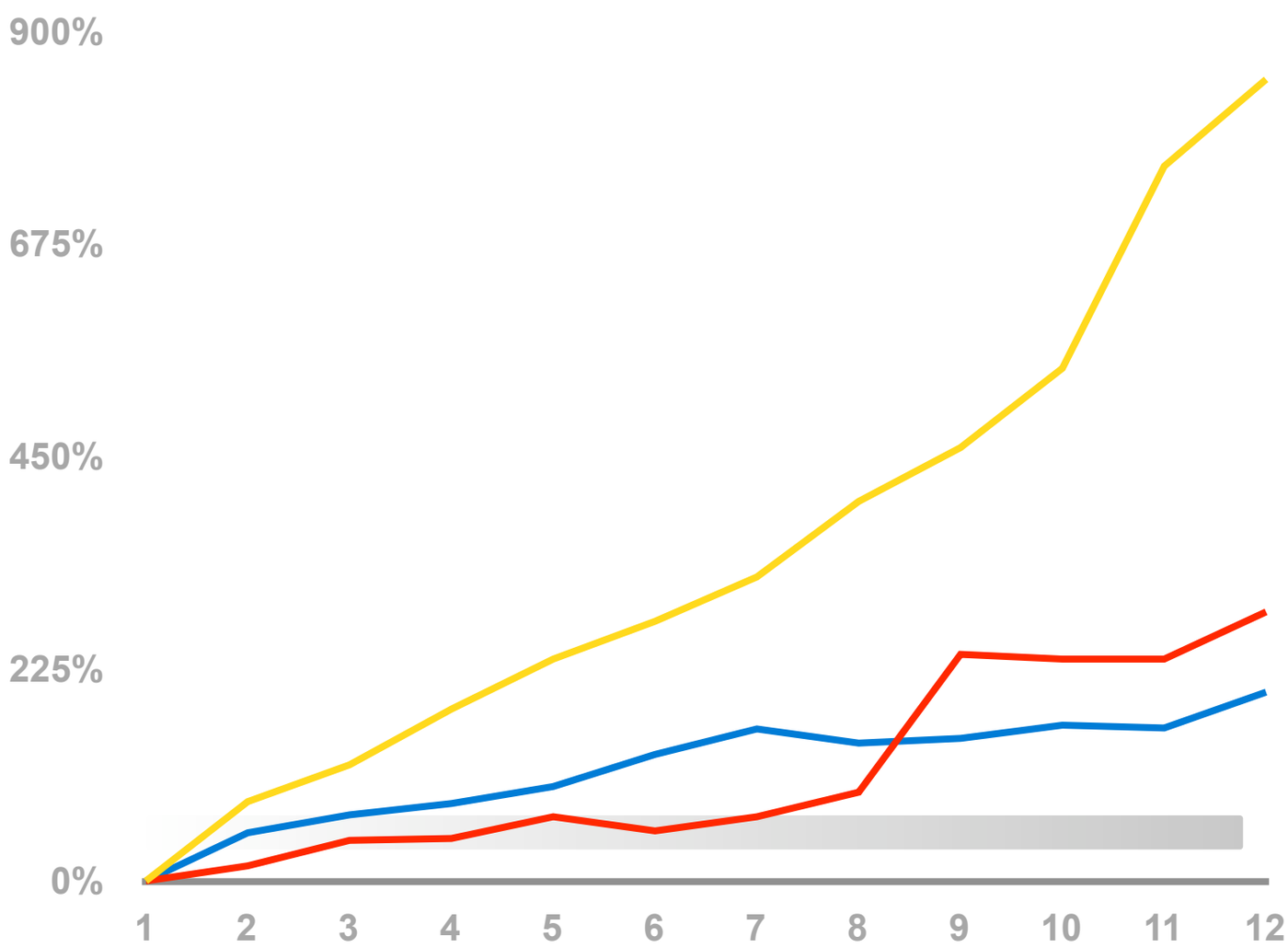


- Chart is positioned to the left of the text box aligned to left guide
- Depending on size of chart and amount of text content, positioning of text box may vary
- Font size can be decreased

Source: Insight Express Survey of 1000 home Internet users (June 2004)

Line Chart Example

Chart title or subtitle placeholder



- Chart is positioned to the left of the text box aligned to left guide
- Depending on size of chart and amount of text content, positioning of text box may vary
- Font size can be decreased

Source: Insight Express Survey of 1000 home Internet users (June 2004)

Matrix Example

Chart title or subtitle placeholder

	Attract Customers	Build Customer Loyalty	Develop New Revenue Streams	Monetize Your Digital Assets
WebSearch	✓	✓		
AdSense for Search			✓	✓
AdSense for Content				✓

Segue Slide

“The perfect search engine would understand
exactly what you mean
and give back exactly what you want.”

Larry Page, Google Co-Founder