

# **Systems Architecture is not Network Topology: Connecting the Consumer Device**

**Bill Foote**

**Jill Chen**

**Ha Nguyen**

**Warner Bros. Advanced Digital Services**

# Talk Outline

- Consumer Devices
  - Why is studying Blu-ray interesting?
- What is Systems Architecture?
- Where should complexity live?
- Long live APIs!
- Case Studies
- How long will this be relevant?

# Why is Blu-ray Interesting?

(to a software engineer)

# Why is Blu-ray interesting?

There are a lot  
of players

- Millions of Blu-ray players
- DVD is ubiquitous

**Walmart**  Christmas costs less at Walmart.

**Panasonic** \$788  
Panasonic® Plasma HDTV 46" diagonal screen, TC-P46U1

**HP** \$298  
HP® 250GB PC, 15.6" diagonal display, HPG60-519WM (minimum 10 per store)

**SONY** \$148  
Sony® Blu-Ray Disc™ Player, 508268 (minimum 10 per store)

**XBOX** \$199  
Xbox 360™ Arcade Console with \$100 gift card. (minimum 10 per store)

**MAGNAVOX** \$29  
Magnavox® DVD player with 1080p upconverter, 545538

**SHARP** \$498  
Sharp® LCD HDTV 42" diagonal screen, LC42SB45UT (not available in all stores)

**One Day In-Store Specials** [LEARN MORE ▶](#)

**Saturday Nov. 7th 8:00 am**

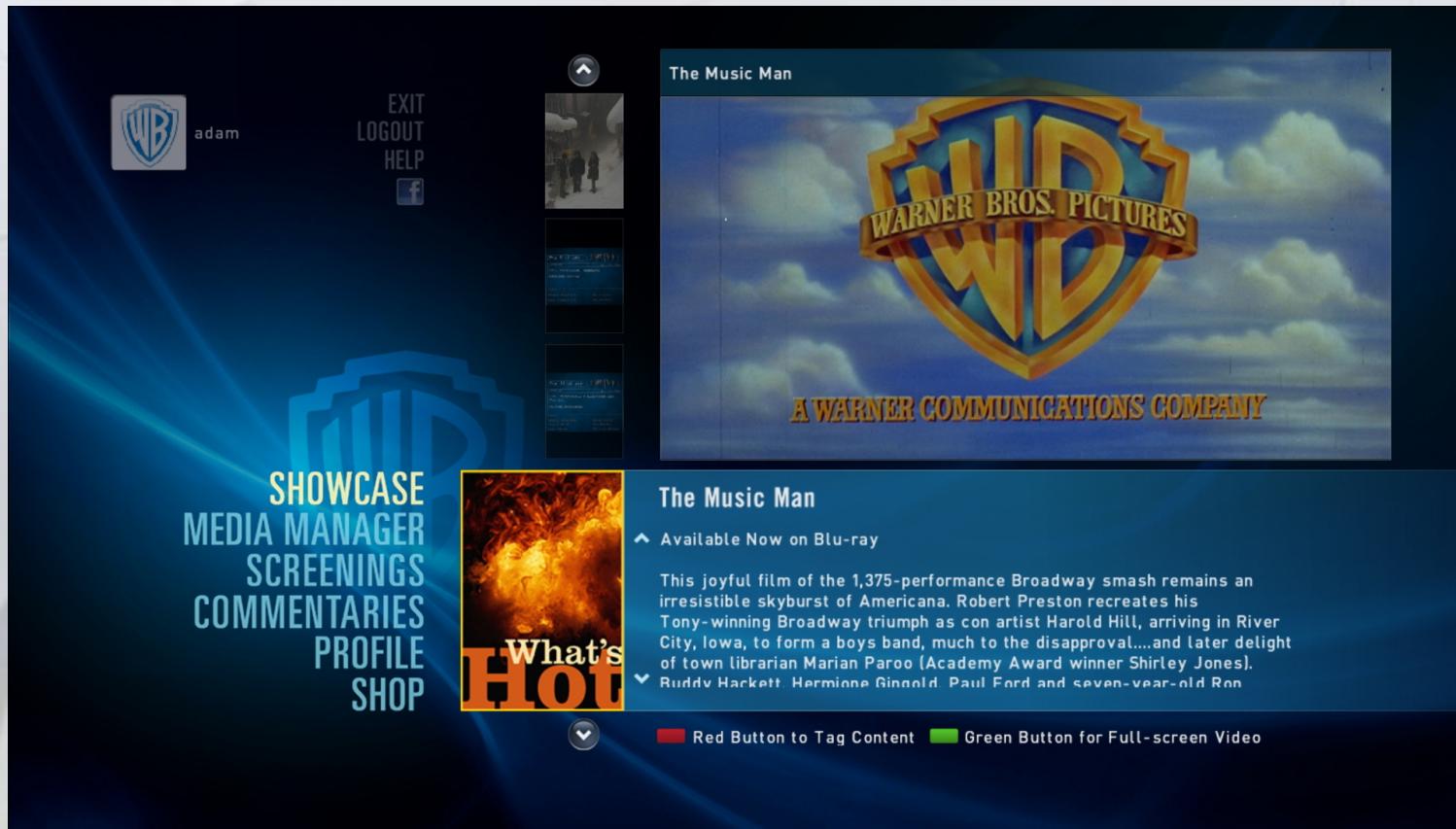


 Print 31069 [Find a Store](#)

Quantities of all items are limited.  
No rainchecks. Prices and availability  
may vary in AK, HI, OK and WI. Not valid  
in Puerto Rico or on Walmart.com.

# Why is Blu-ray interesting?

Players connect to the Internet



# Why is Blu-ray interesting?

They're on  
the best  
screen in  
the house

- best audio,  
too!
- in a social  
space



# Why is Blu-ray interesting?

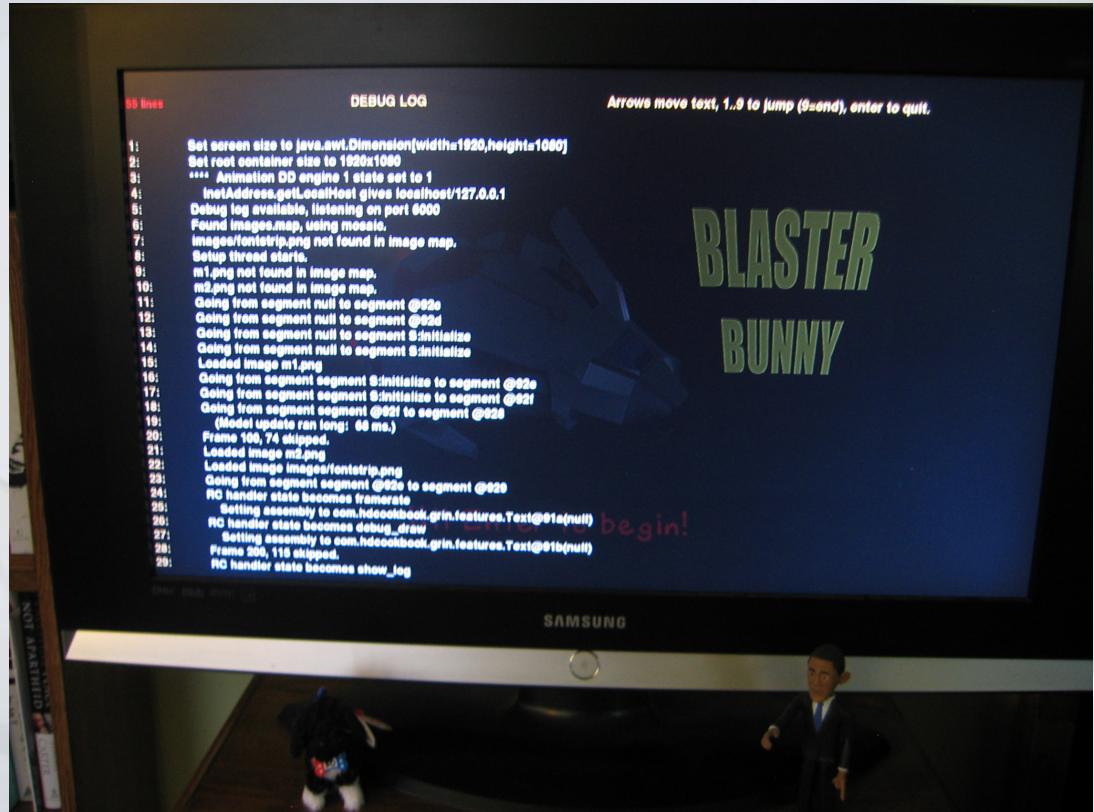
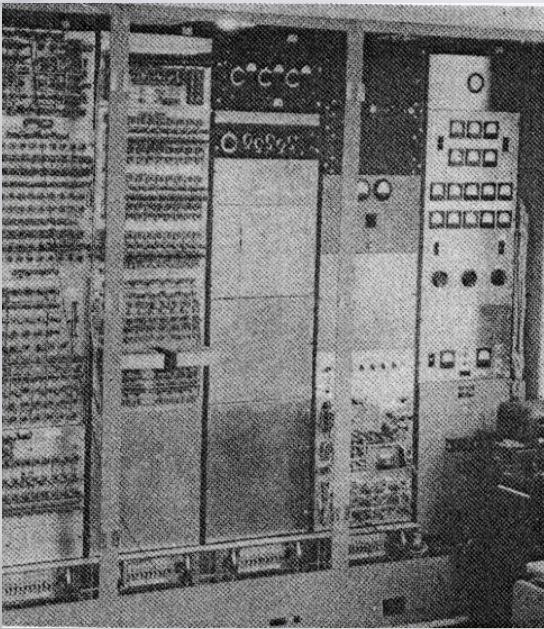
Consumer  
players have  
a slow CPU\*  
and limited  
memory

(\*) except game  
consoles



# Why is Blu-ray interesting?

Debugging  
tools are  
limited



# Why is Blu-ray interesting?

“Best Practices” from PC, mobile phone  
don’t always apply.

*“what are the scarce resources being  
optimized?”*

# Why is Blu-ray interesting?

BUT, many lessons learned will continue to apply in the living room and elsewhere, for a long time to come.

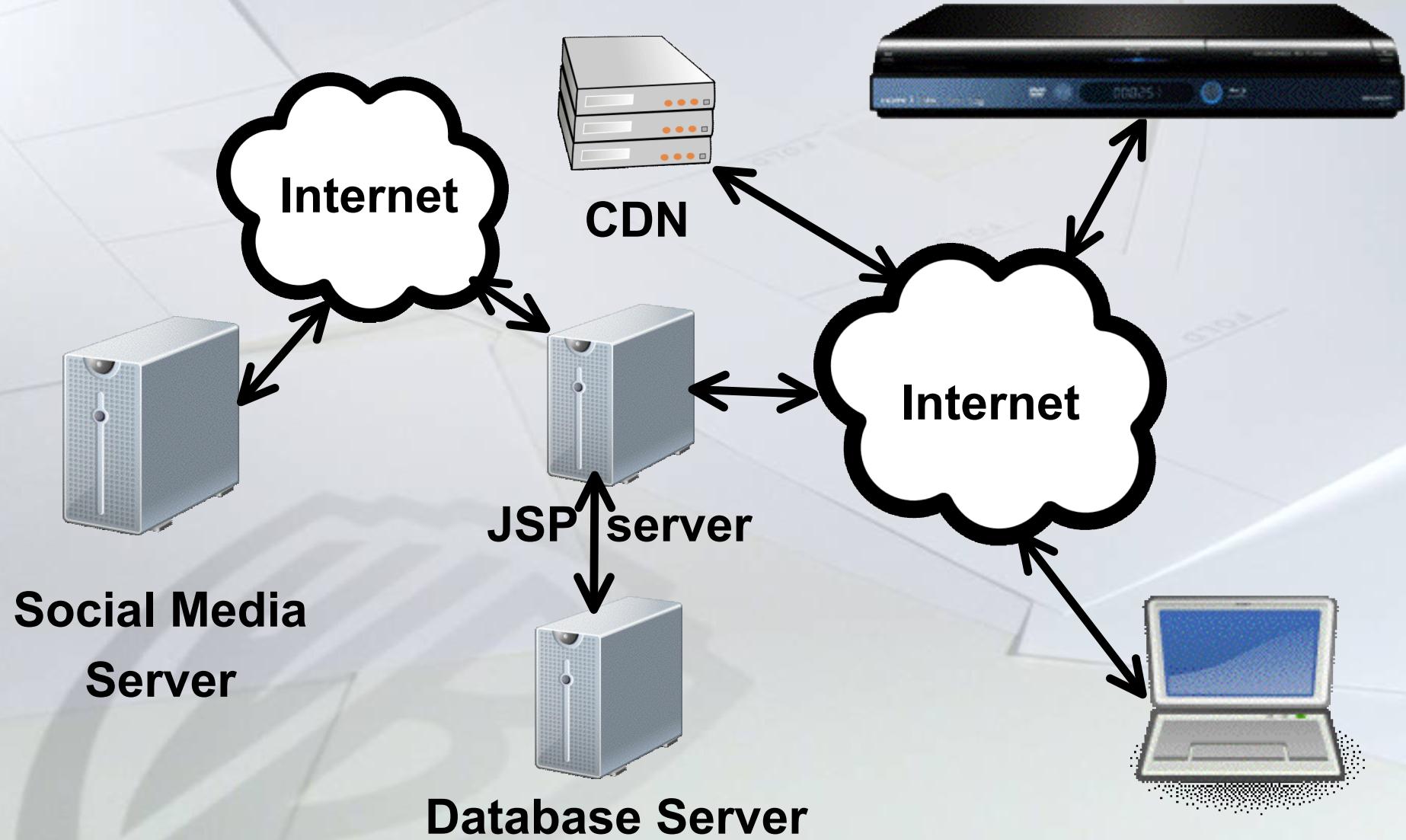
# **What is Systems Architecture?**

# What is Systems Architecture?

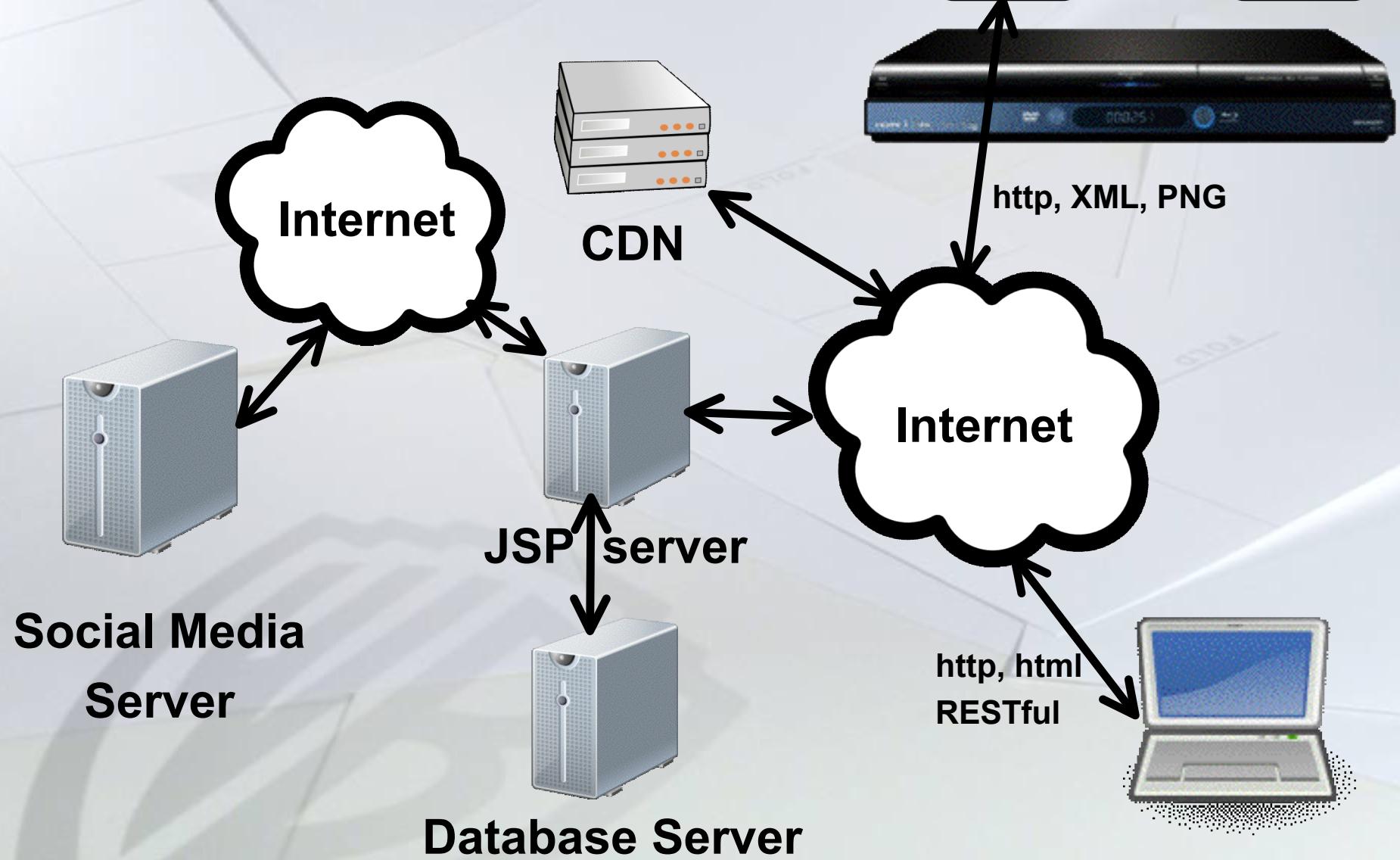
“The conceptual design that defines the structure and/or behavior of a system”  
(Wikipedia)

“The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.”  
(ANSI-IEEE 1471-2000)

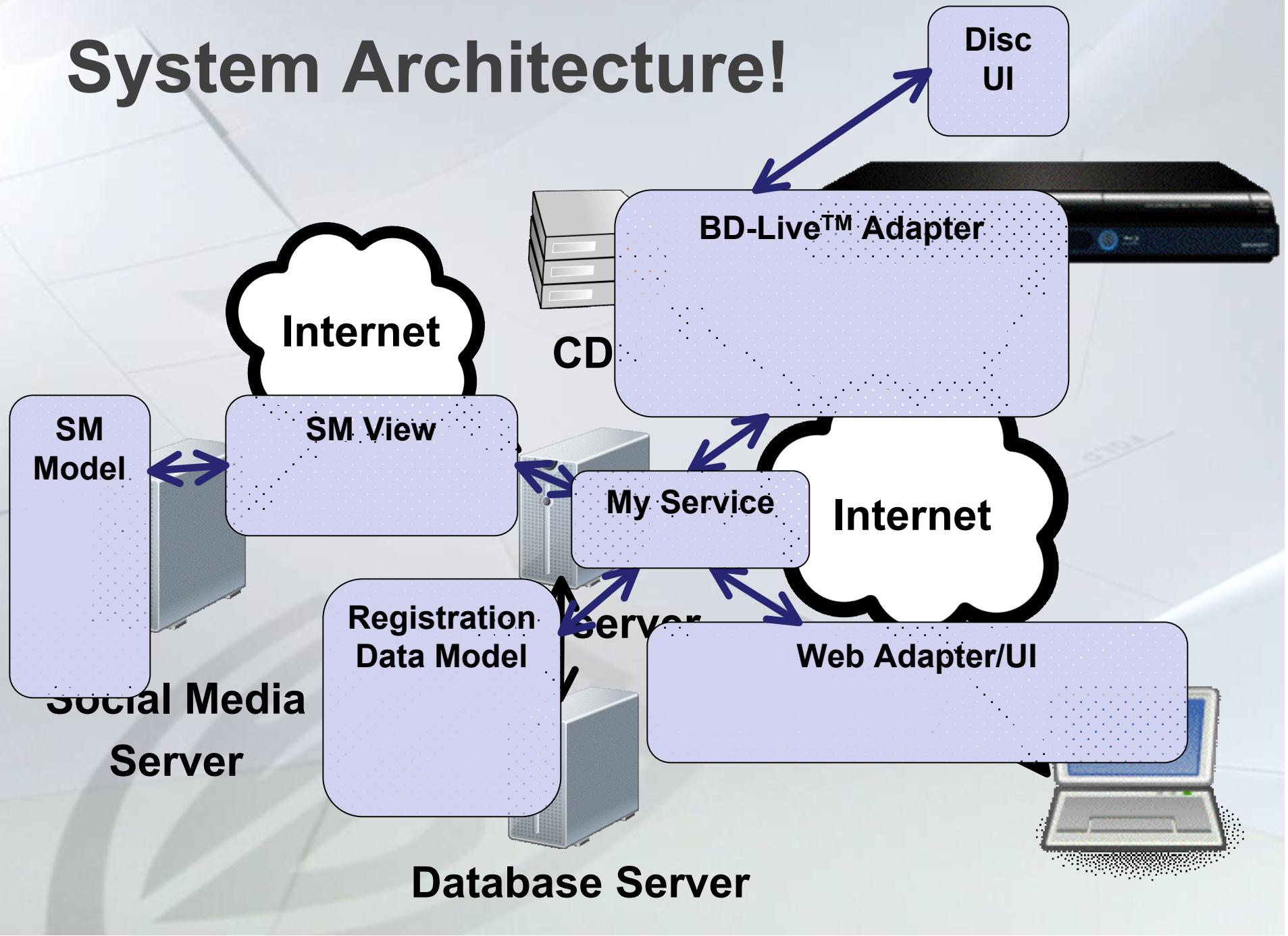
# Network Topology



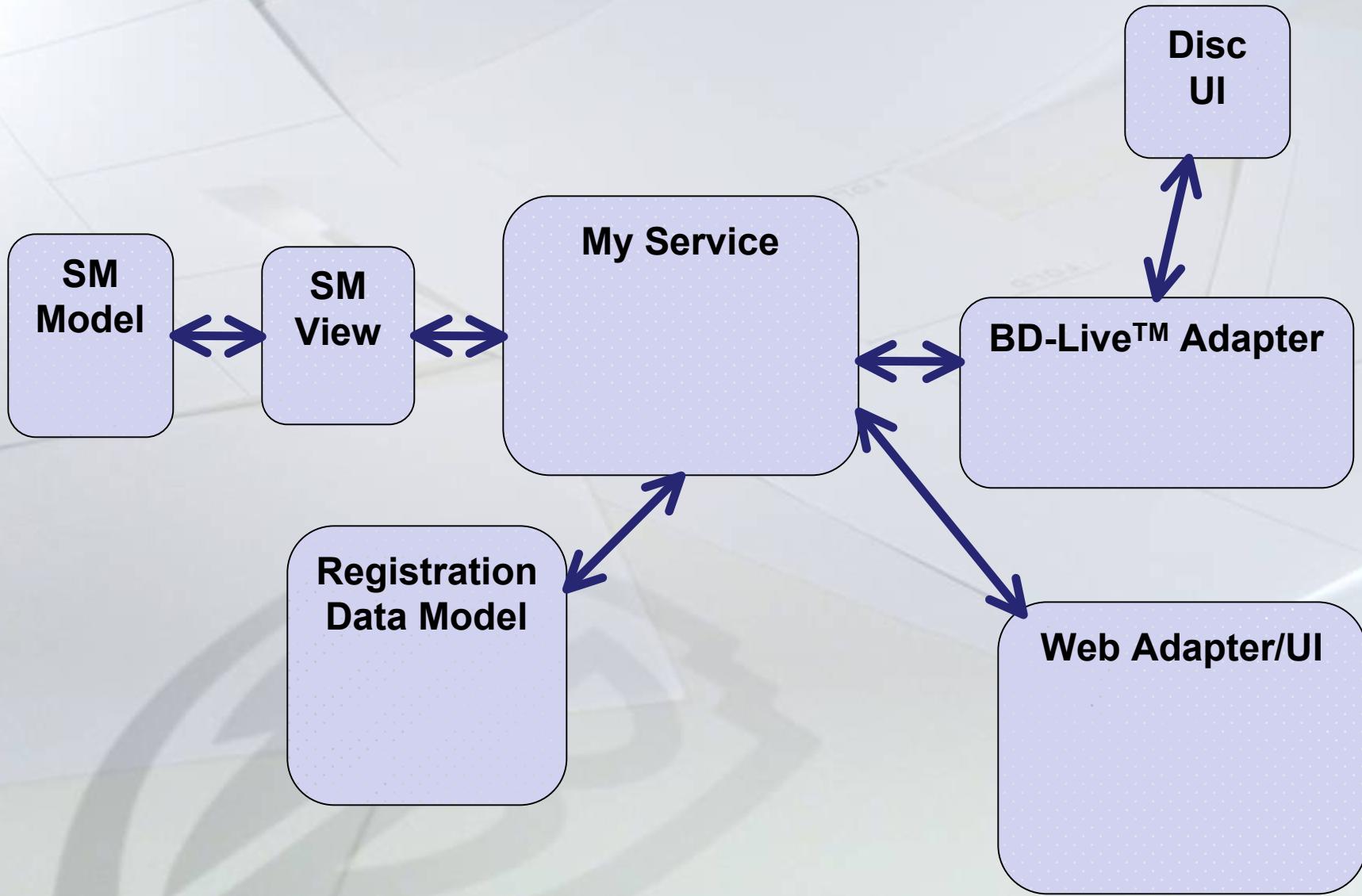
# System Architecture?



# System Architecture!



# Systems Architecture!



# Systems Architecture!

*“... principles governing its design and evolution”*

- “Strenuously Avoid VFS updates”
  - “Whenever practical, put business logic on the server”
- “Minimize server round trips”
  - a round trip costs 30-500 ms!
- “Push content to the CDN when possible”
- “In BD-J code, minimize the number of classes, because class loading is slow”
- “Measure launch time performance, particularly after initial disc boot”

# Systems Architecture

A system's architecture isn't a diagram.

A diagram is just *one view* onto a portion  
of a system's architecture.

# Systems Architecture

## Conway's Law:

“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.”

Conway, Melvin E. (April, 1968), "How do Committees Invent?", Datamation 14 (5): 28–31,  
[http://www.melconway.com/Home/Conways\\_Law.html](http://www.melconway.com/Home/Conways_Law.html)

# Where Should Complexity Live?

# Where should complexity live?

Developing code that runs on a living room device is expensive.

# Where should complexity live?

Developing code that runs on a living room device is expensive.

Really expensive.

# Where should complexity live?

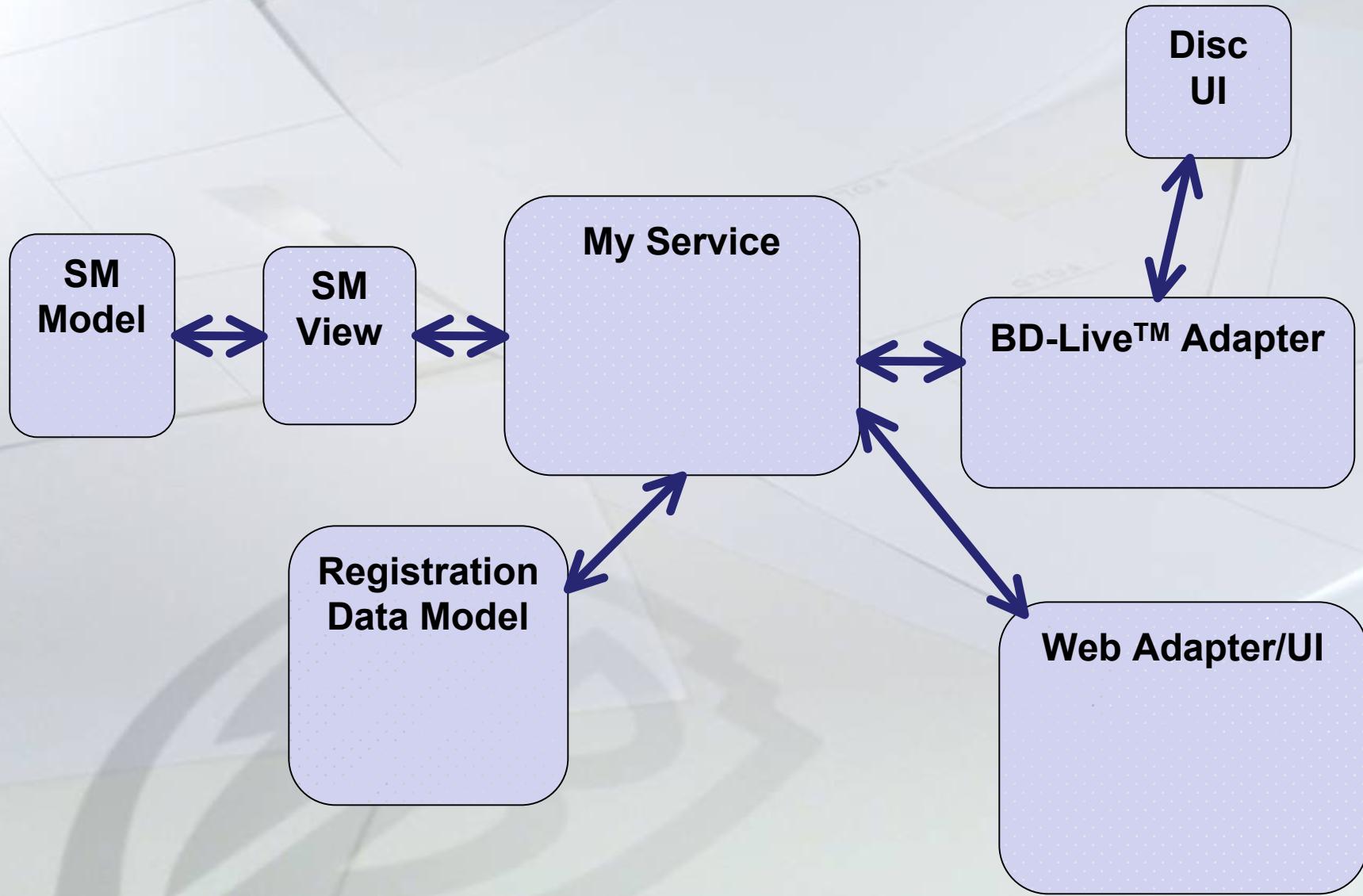
Developing code that runs on a living room device is expensive.

Really expensive.

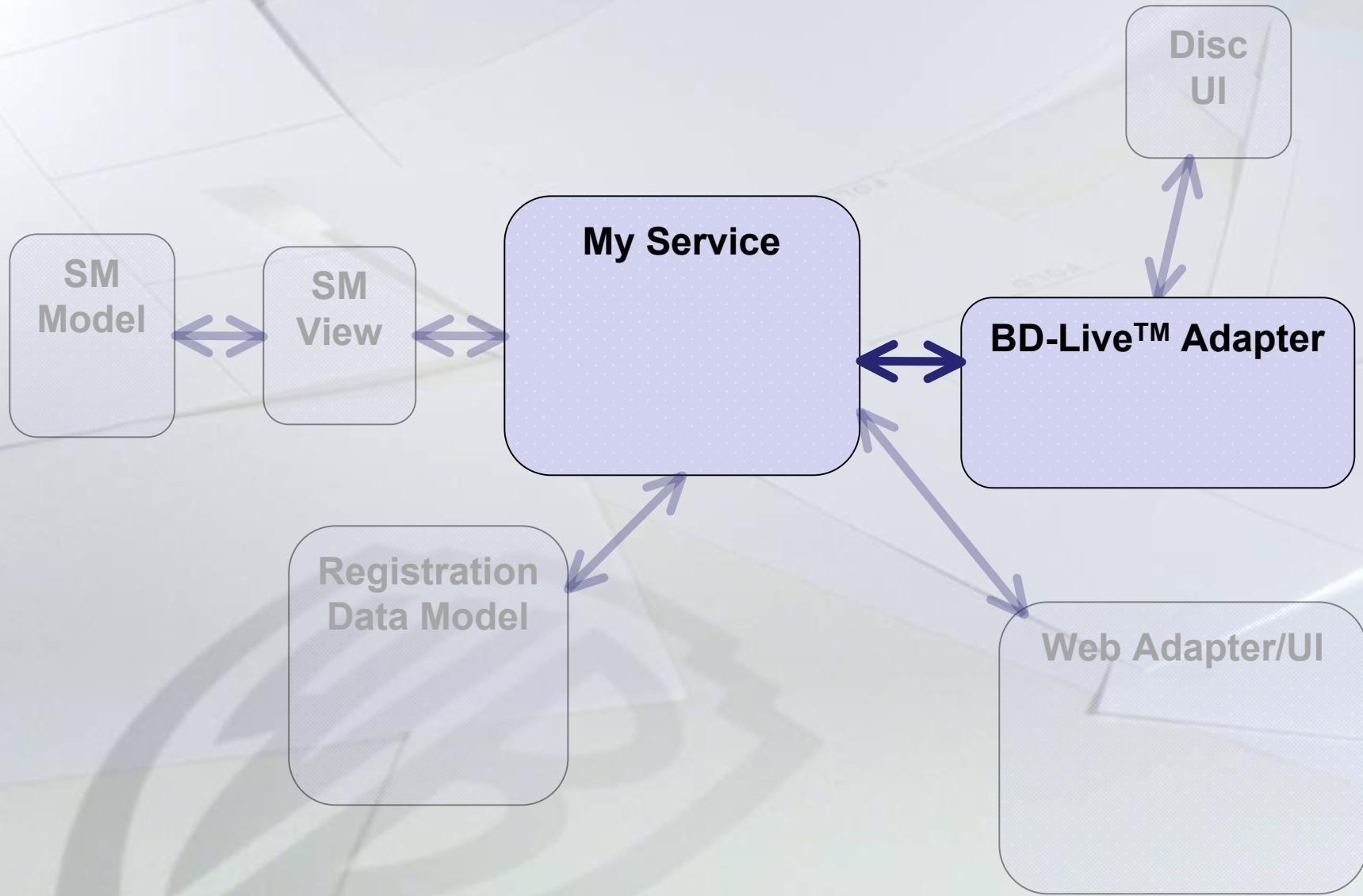
Why?

# **Discussion of BD-J development challenges**

# Where should complexity live?

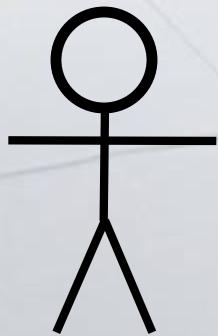


# Where should complexity live?

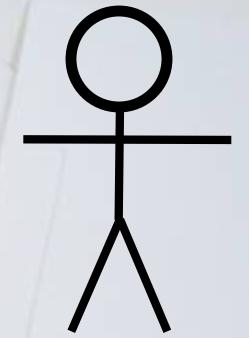
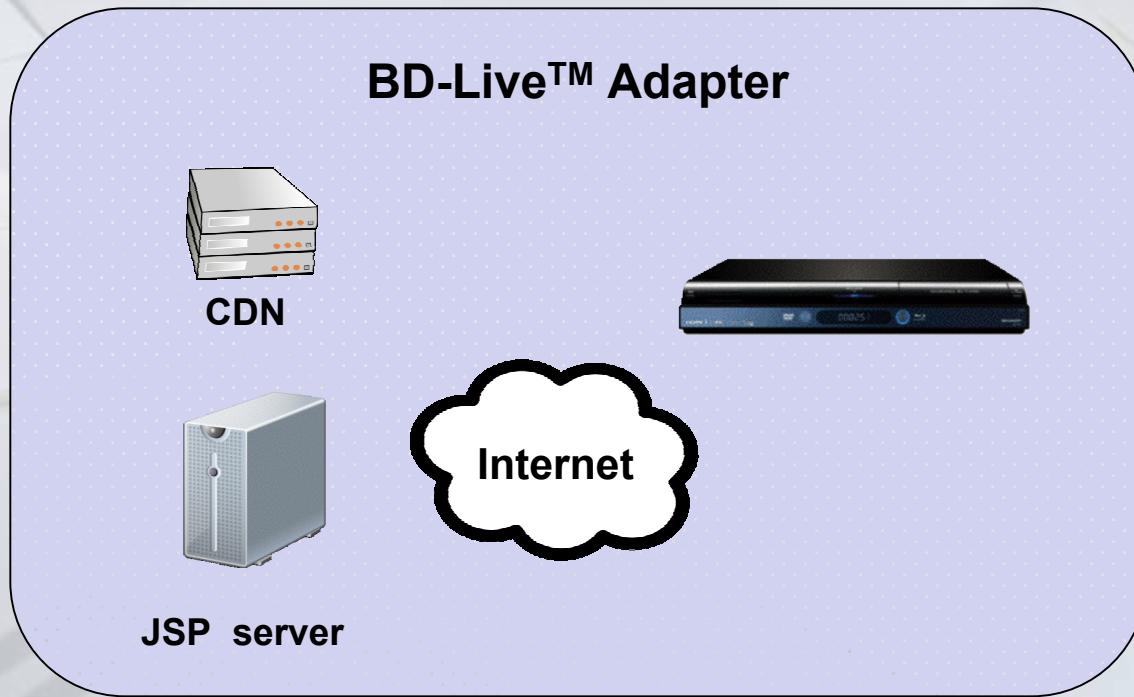




# Where should complexity live?

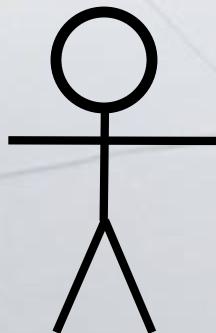


**My Service**

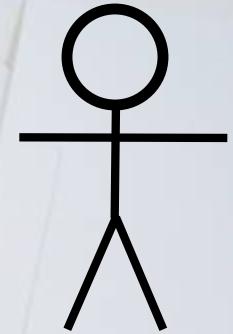
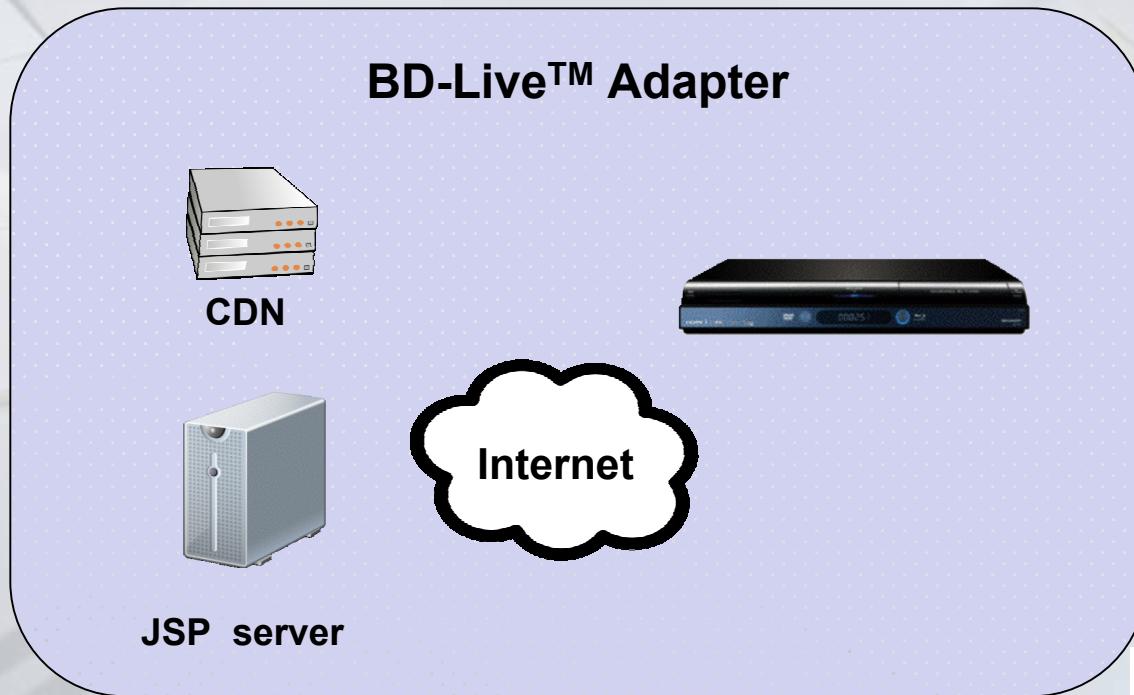


**Disc UI**

# Where should complexity live?



**My Service**

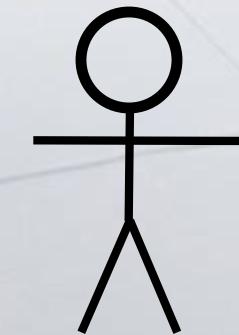


**Disc UI**

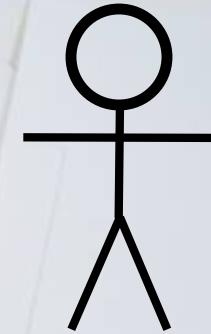
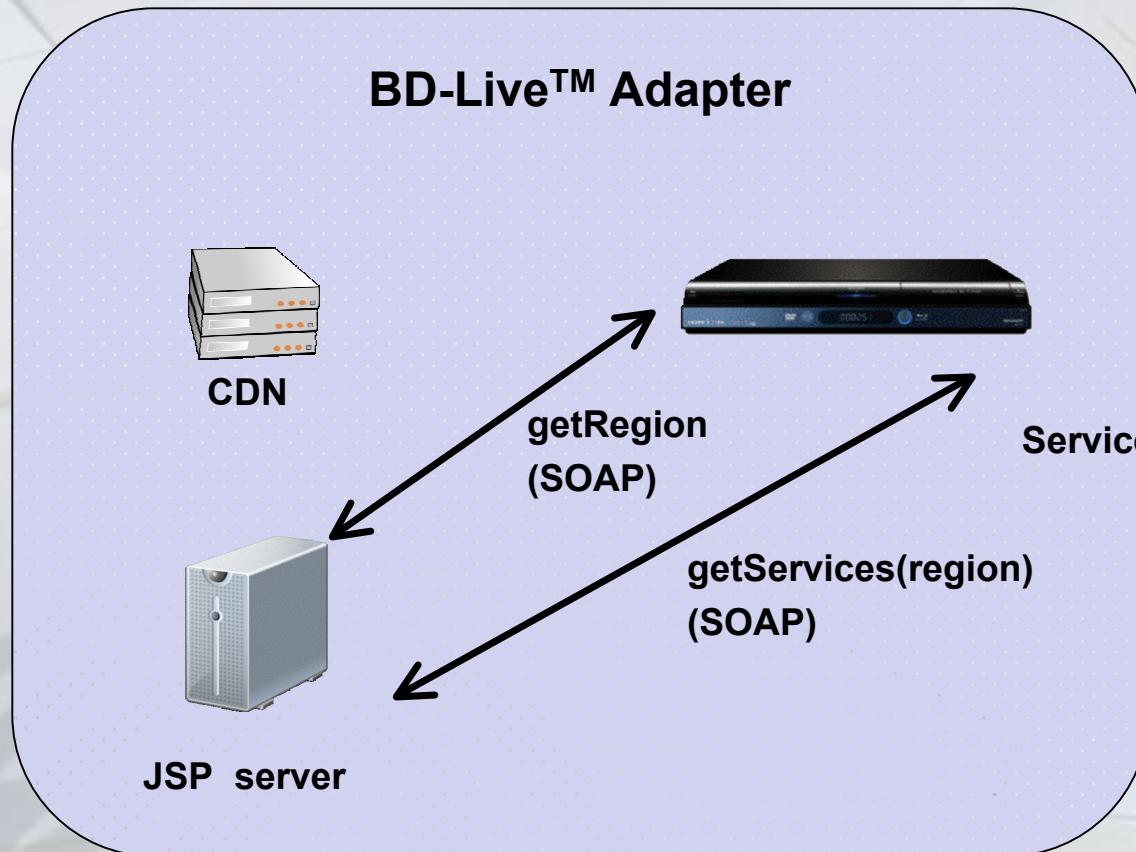


**Developers**

# Where should complexity live?



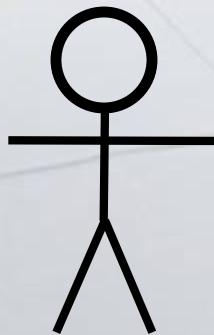
**My Service**



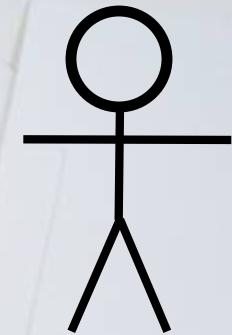
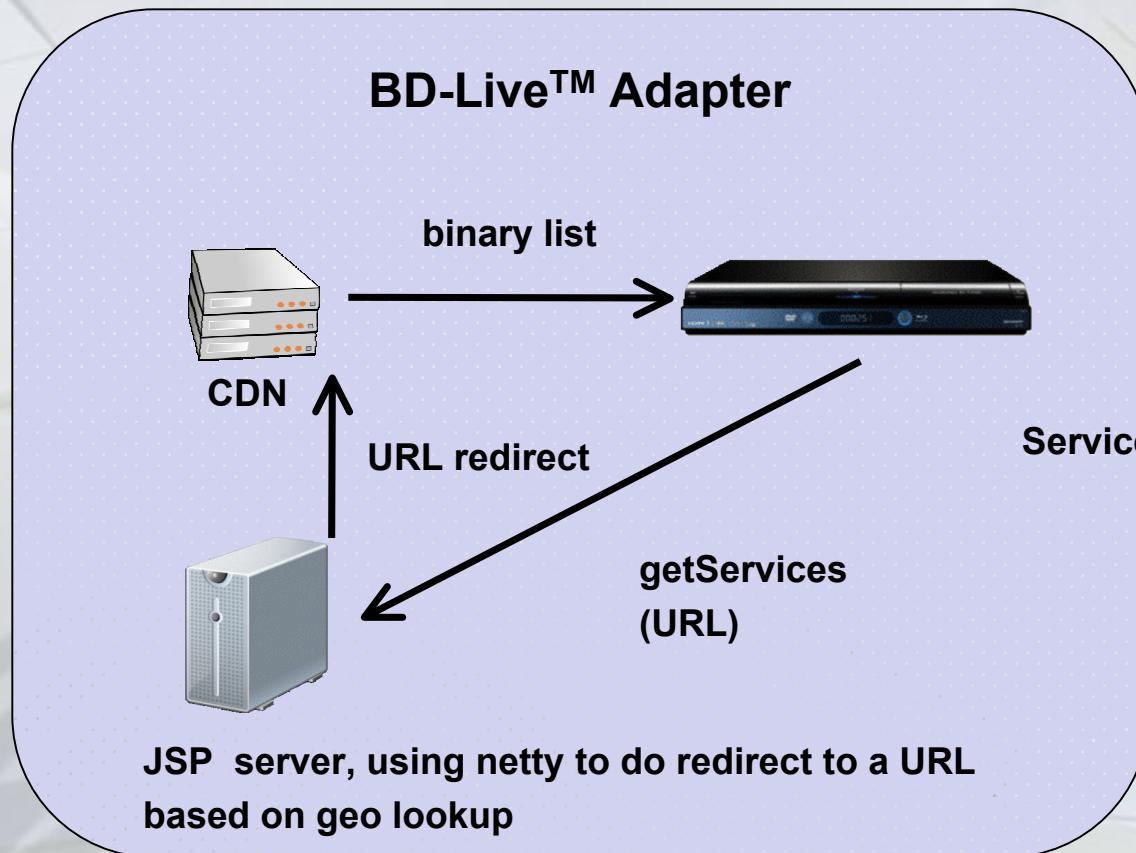
**Disc UI**

**getting a list of services for your region**

# Where should complexity live?

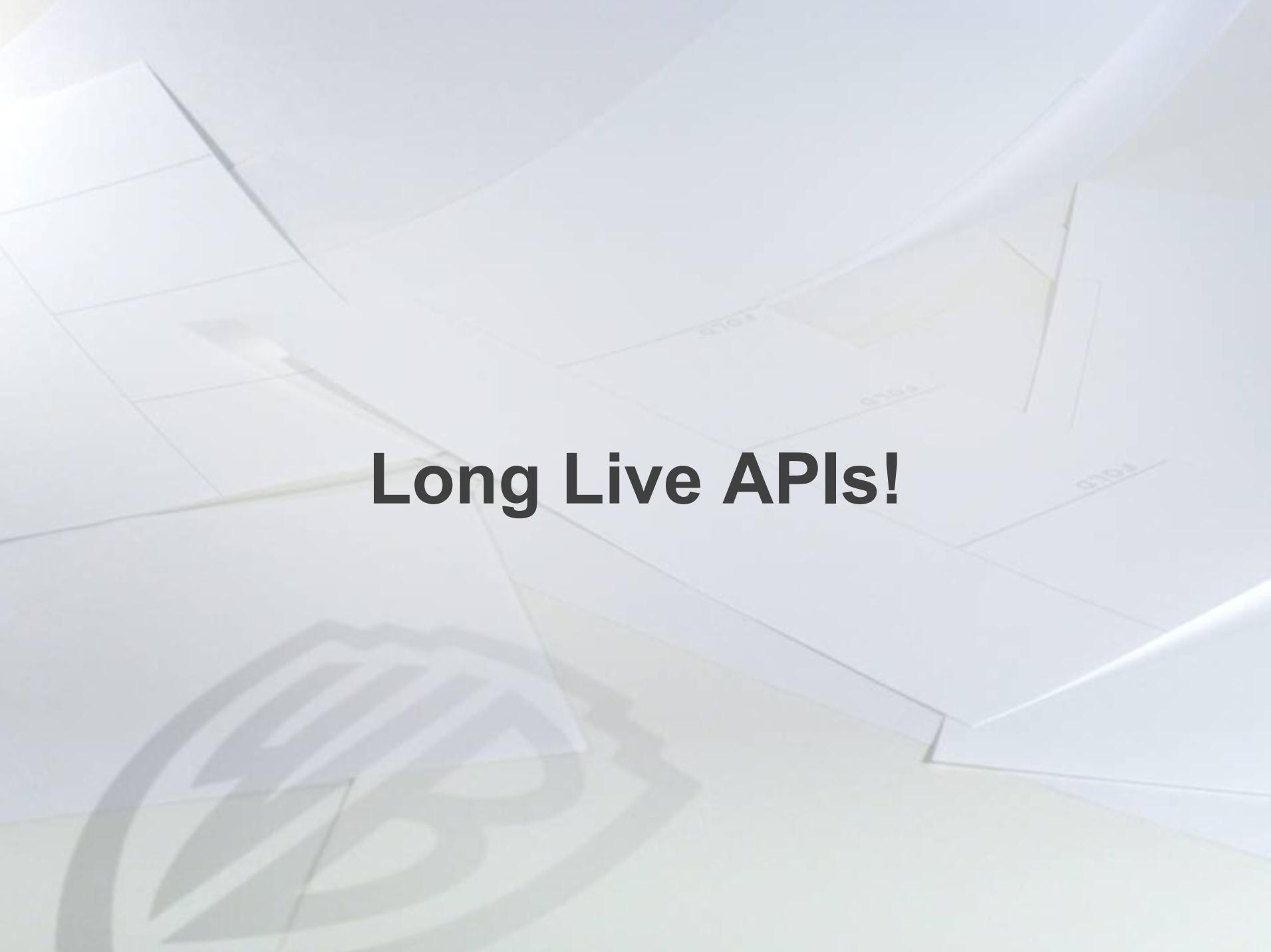


**My Service**



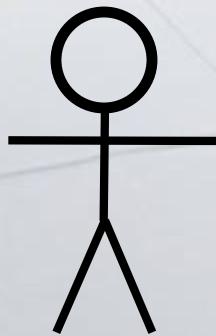
**Disc UI**

**getting a list of services for your region**

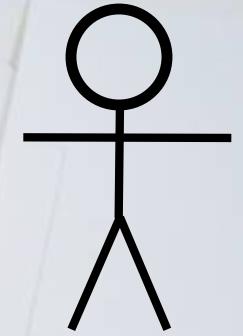
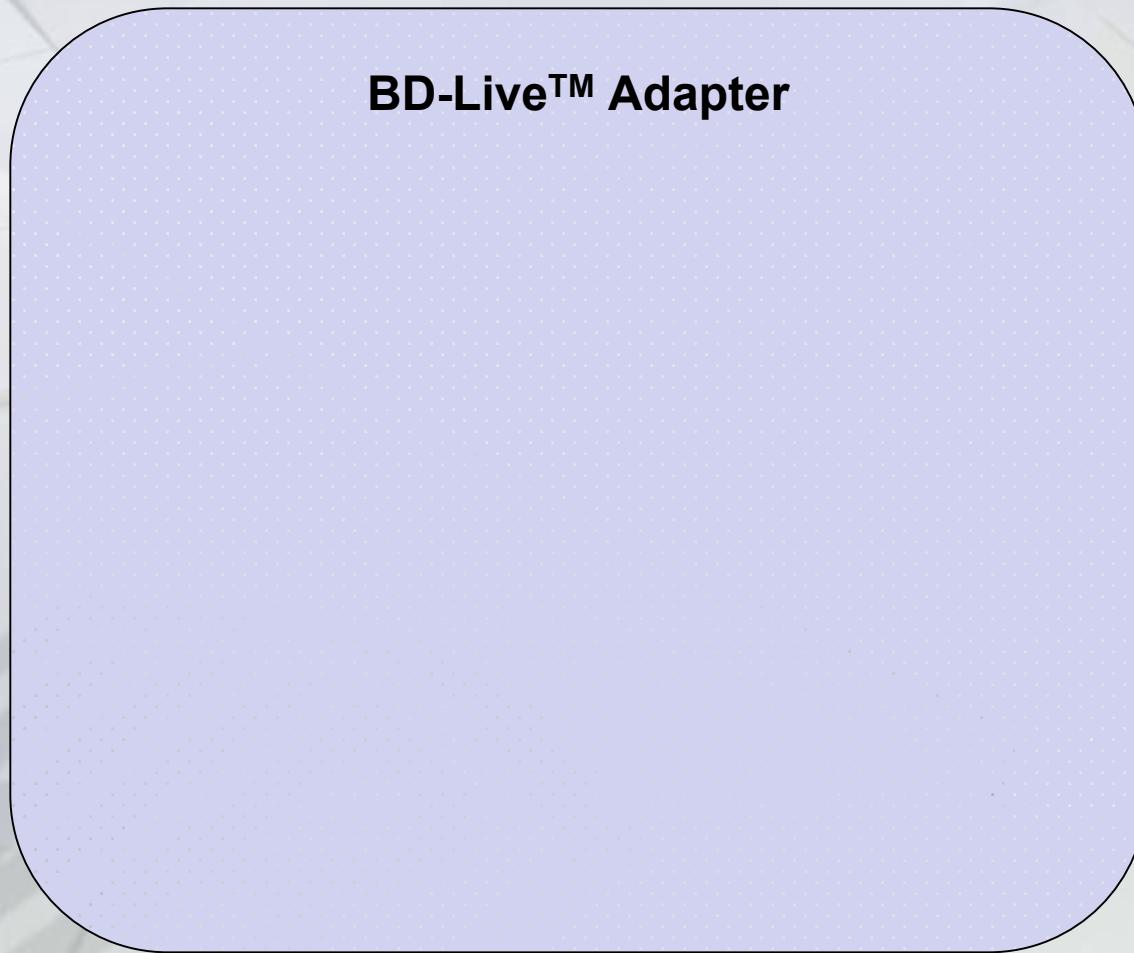


**Long Live APIs!**

# Long live APIs!

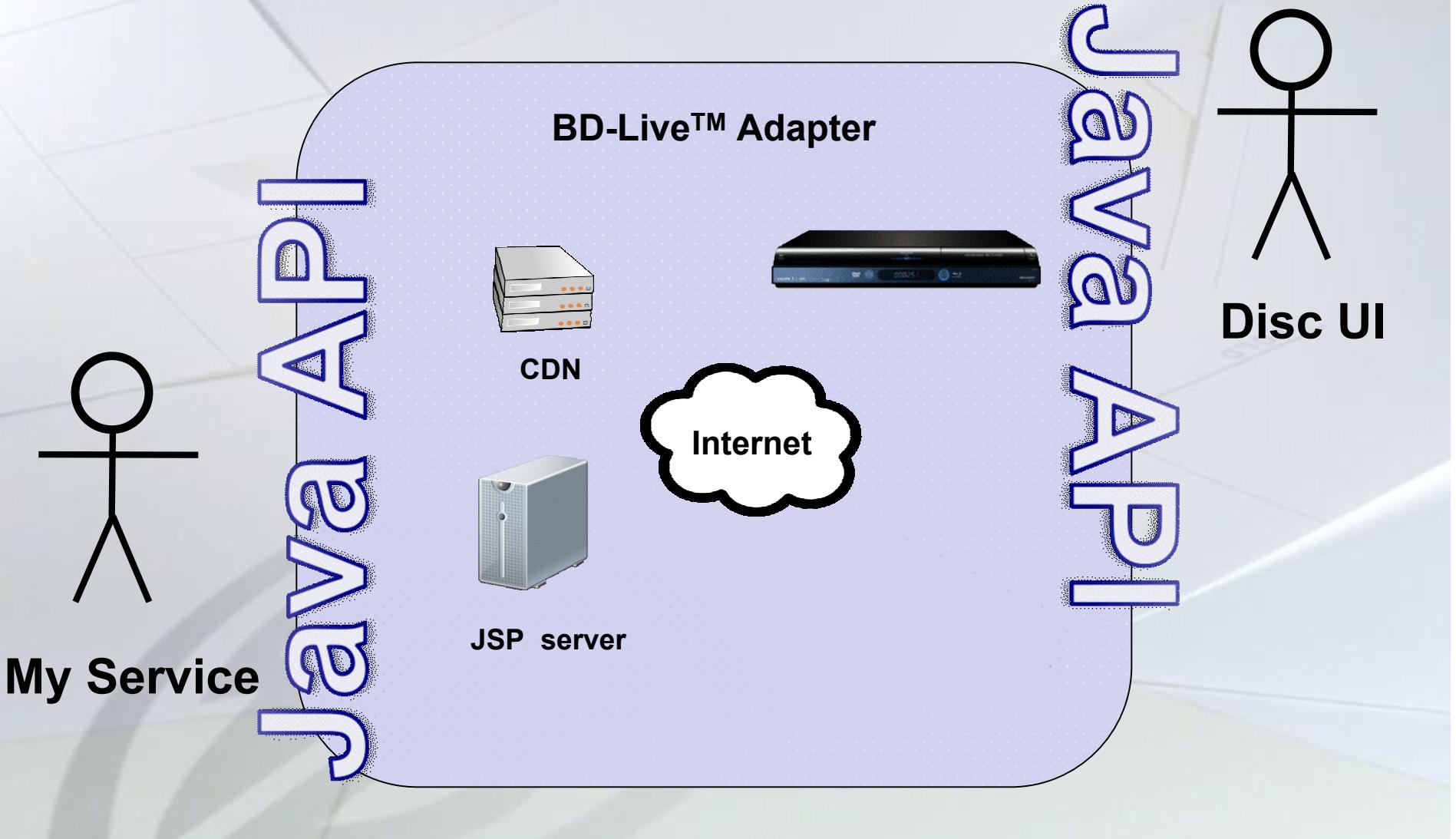


**My Service**



**Disc UI**

# Long live APIs!



# Long live APIs!

An API can be

- Classes and Methods
  - usually with rich data structures
- Attribute/Value pairs
- A network protocol (like SOAP or RMI)

... but ultimately, your customer needs  
classes and methods anyway!

# Case Study

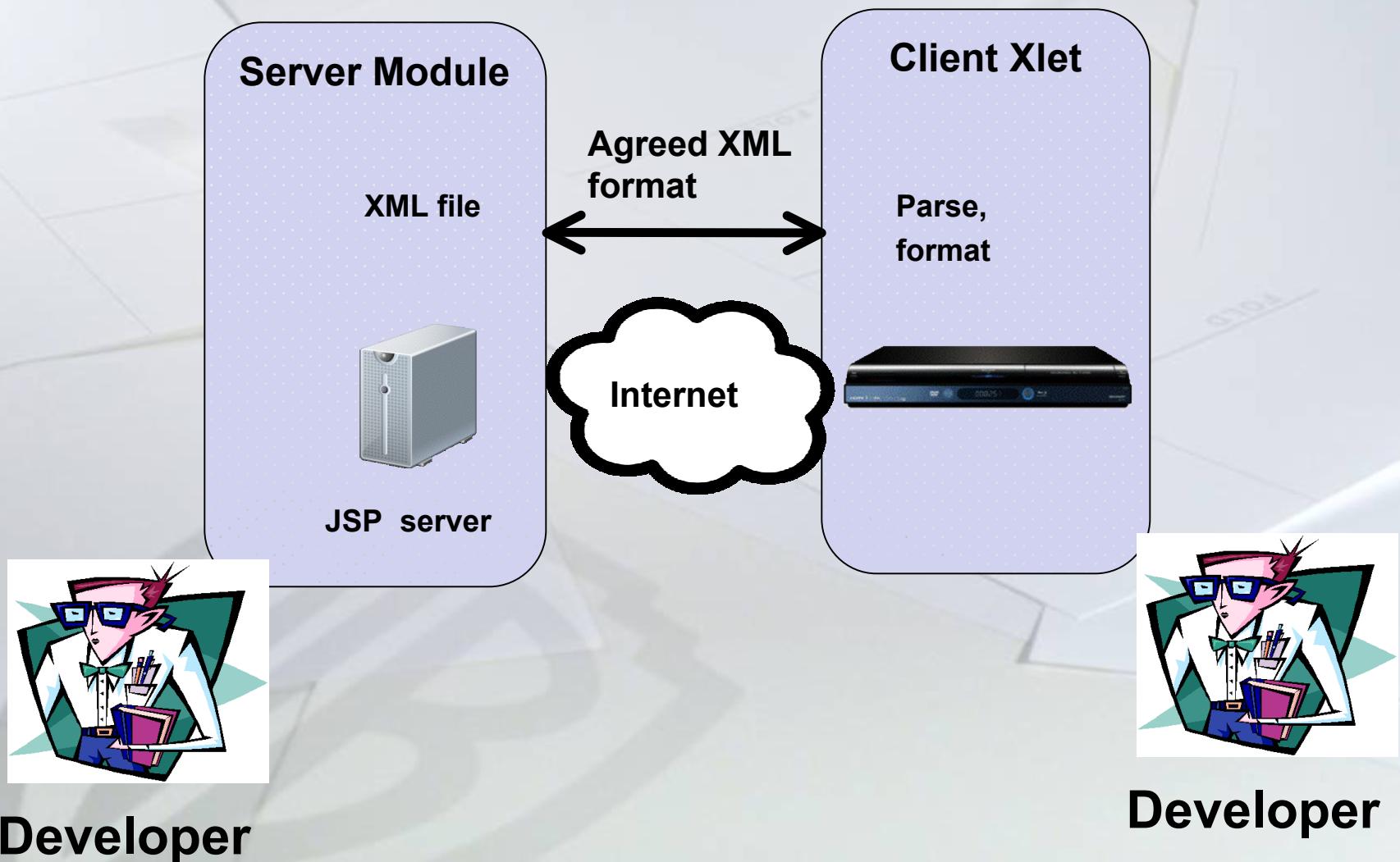
# Case Study: 50K of structured data

Among the stories of world-wide renown, not the least stirring are those that have gathered about the names of national heroes. The neid, the Nibelungenlied, the Chanson de Roland, the Morte D'Arthur, they are not history, but they have been as National Anthems to the races, and their magic is not yet dead.

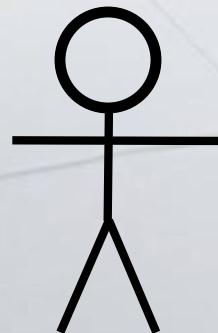
In olden times our forefathers used to say that the world had seen nine great heroes, three heathen, three Jewish, and three Christian; among

• • •

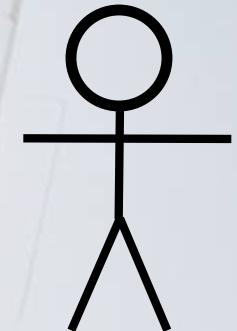
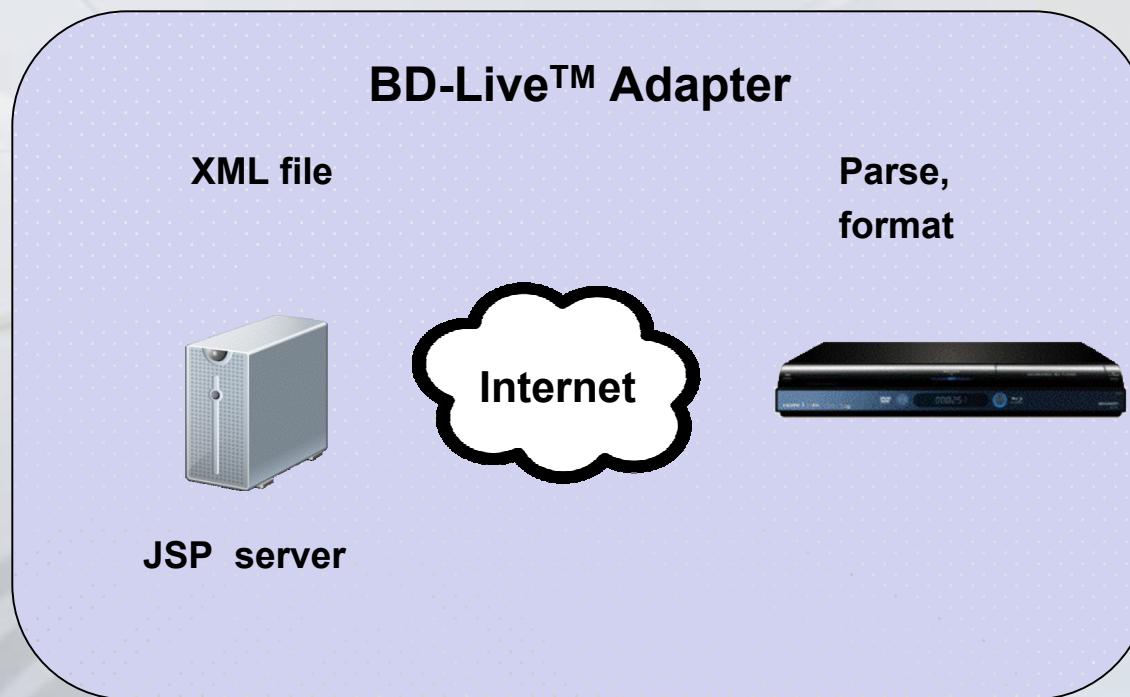
# Prototype Implementation Topology-based Architecture



# Prototype Implementation Adapter-centric Architecture



**My Service**

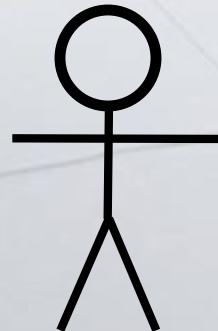


**Disc UI**

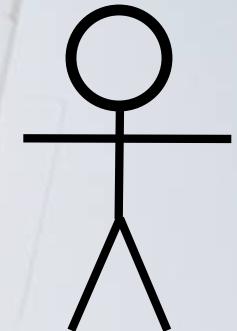
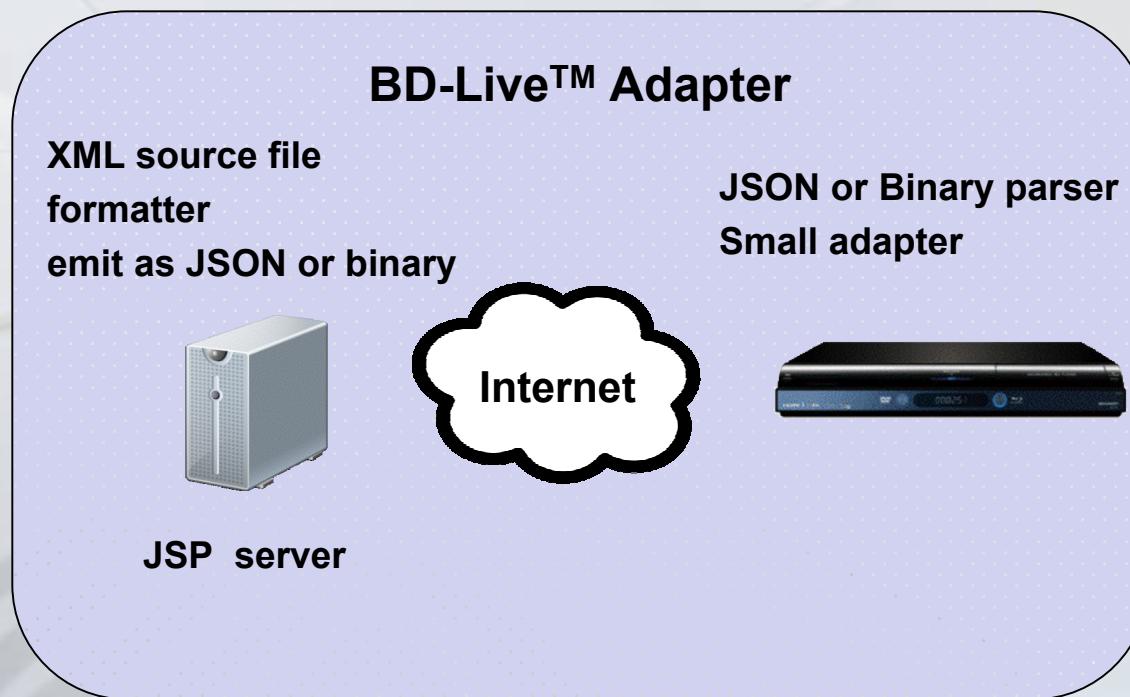


**Developer**

# Optimized Implementation Adapter-centric Architecture



**My Service**

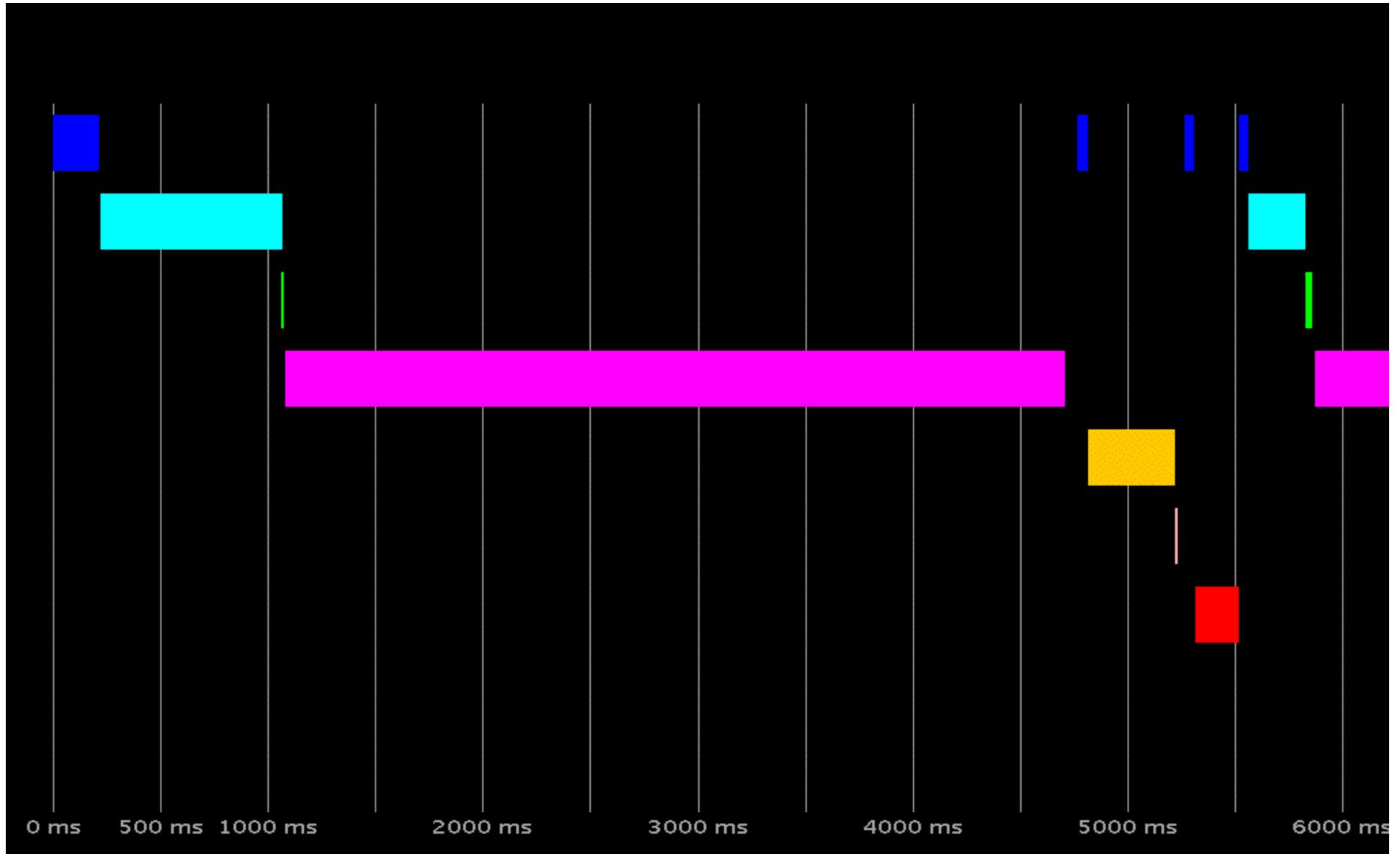


**Disc UI**



**Developer**

# Modularization Impedes Optimization?



blue    Open URL  
cyan    Read XML  
green    Unpack XML

magenta    Format XML  
orange    Read JSON  
pink    Unpack JSON

red    Read Binary

# Open URL

```
if (MyDirector.PROFILE) {  
    token = Profile.startTimer(profileOpenURL, TID);  
}  
  
URL url = new URL(  
    "http://hdcookbook.com/tmp/morte_darthur.xml");  
InputStream is = url.openStream();  
Reader r = new InputStreamReader(  
    new BufferedInputStream(is), "UTF-8");  
  
if (MyDirector.PROFILE) {  
    Profile.stopTimer(token);  
}
```

# Open URL

start 0.0ms for **216.06499ms**

start 4764.651ms for **51.204ms**

start 5267.998ms for **42.6ms**

start 5518.395ms for **43.239ms**

start 9215.803ms for **43.903ms**

start 9512.844ms for **39.125ms**

...

# Read XML

```
if (MyDirector.PROFILE) {  
    token = Profile.startTimer(profileReadXML, TID);  
}  
  
Parser parser = new Parser();  
Element e = parser.parse(r);  
r.close();  
  
if (MyDirector.PROFILE) {  
    Profile.stopTimer(token);  
}
```

# Read XML

start 218.234ms for **845.008ms**

start 5563.4453ms for **262.75ms**

start 9747.59ms for **277.831ms**

start 13939.895ms for **252.868ms**

start 17978.623ms for **279.3ms**

# Unpack XML

```
ArrayList chapList = new ArrayList();
visitXMLGraph(e, chapList);
```

# Unpack XML

start 1065.226ms for **15.522ms**

start 5828.14ms for **33.351ms**

start 10031.916ms for **7.358ms**

start 14193.805ms for **7.583ms**

start 18258.996ms for **4.635ms**

# Format XML

```
String line = "";
for (;;) {                                // build line
    int pos = remaining.indexOf(' ', 1);
    if (pos == -1) {
        pos = remaining.length();
    }
    String word = remaining.substring(0, pos);
    String newLine = line + word;
    if (font.getStringWidth(newLine) < MAX_WIDTH) {
        line = newLine;
        remaining = remaining.substring(pos);
    }
    ...
}
```

# Format XML

start 1083.978ms for **3627.935ms**

start 5872.732ms for **3341.481ms**

start 10040.715ms for **3303.34ms**

start 14202.37ms for **3268.1228ms**

start 18264.861ms for **3290.157ms**

# Read JSON

```
Object[] book = (Object[]) JsonIO.readJSON(rdr);  
rdr.close();
```

# Read JSON

start 4817.9453ms for **403.928ms**

start 9261.314ms for **226.835ms**

start 13388.146ms for **240.62099ms**

start 17512.295ms for **224.661ms**

start 21597.291ms for **228.325ms**

# Unpack JSON

```
Object[] strings = (Object[]) book[0];
Object[] text = (Object[]) book[1];
Object[] font = (Object[]) book[2];
Chapter[] result = new Chapter[text.length];
for (int i = 0; i < result.length; i++) {
    Chapter chapter = new Chapter();
    result[i] = chapter;
    Object[] textI = (Object[]) text[i];
    Object[] fontI = (Object[]) font[i];
    chapter.pages = new Page[textI.length];
    for (int j = 0; j < textI.length; j++) {
```

# Unpack JSON

start 5223.747ms for **14.158ms**

start 9489.905ms for **21.257ms**

start 13637.909ms for **2.513ms**

start 17738.334ms for **2.874ms**

start 21826.746ms for **6.52ms**

# Read/Unpack Binary

```
String[] strings =
    new String[dis.readShort() & 0xffff];
for (int i = 0; i < strings.length; i++) {
    strings[i] = dis.readUTF();
}
Chapter[] result
    = new Chapter[dis.readShort() & 0xffff];
for (int i = 0; i < result.length; i++) {
    Chapter chapter = new Chapter();
    result[i] = chapter;
    . . .
```

# Read/Unpack Binary

start 5314.043ms for **202.40099ms**

start 9553.601ms for **150.18701ms**

start 13734.05ms for **150.98499ms**

start 17782.836ms for **149.66301ms**

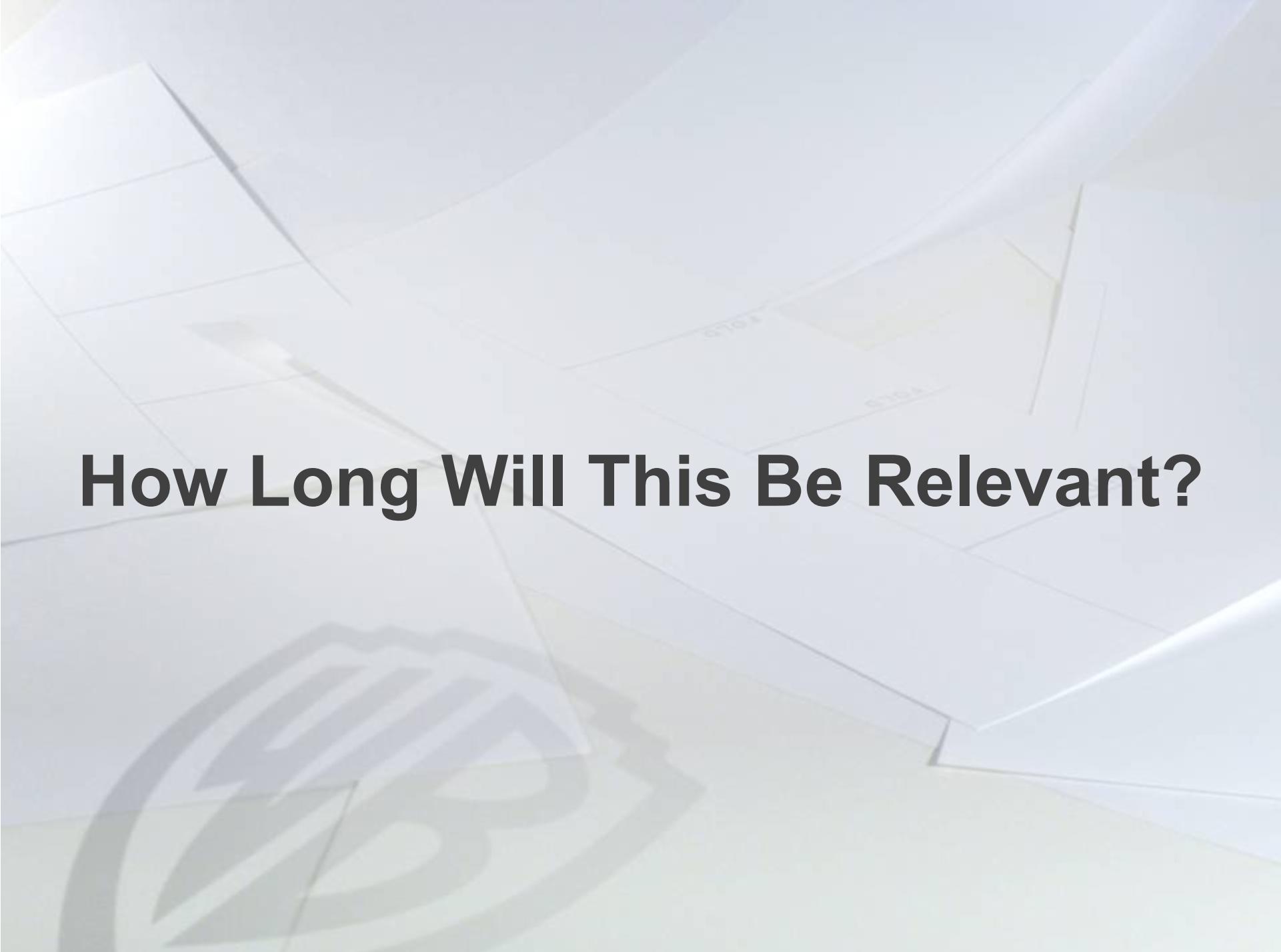
start 21876.705ms for **161.51901ms**

# The Winner is...

**XML:  $50 + 850 + 15 + 3625 = 4540 \text{ ms}$**

**JSON:  $50 + 405 + 15 = 470 \text{ ms}$**

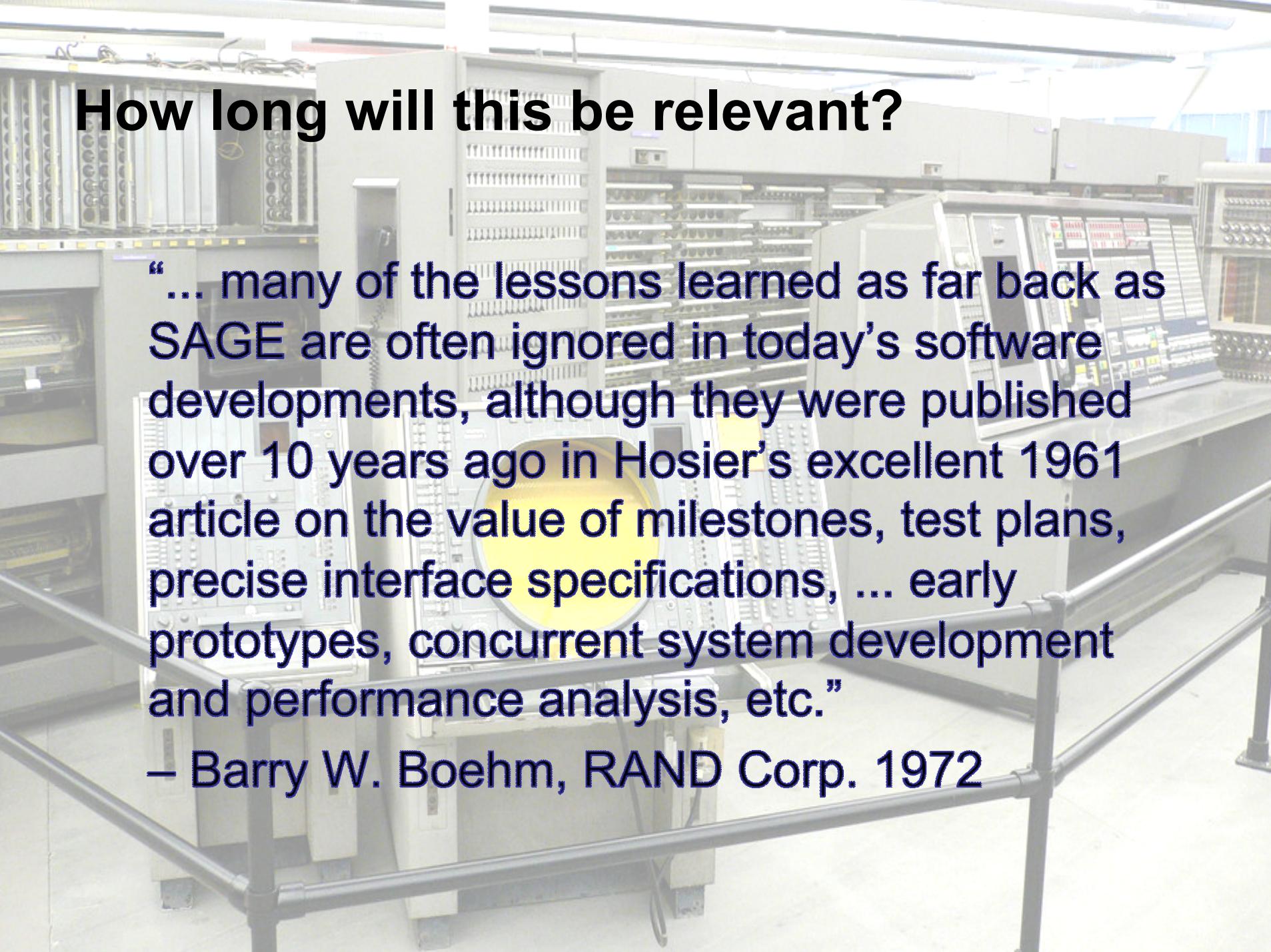
**Binary:  $50 + 200 = 250 \text{ ms}$**

The background of the slide features a light gray abstract design composed of various geometric shapes, including triangles and rectangles. Faint, illegible text is visible through the transparency of the shapes, appearing to read "CHIPS", "CPU", and "GPU".

# **How Long Will This Be Relevant?**

# How long will this be relevant?

“... the value of milestones, test plans, precise interface specifications, ... early prototypes, concurrent system development and performance analysis, etc.”



# How long will this be relevant?

“... many of the lessons learned as far back as SAGE are often ignored in today’s software developments, although they were published over 10 years ago in Hosier’s excellent 1961 article on the value of milestones, test plans, precise interface specifications, ... early prototypes, concurrent system development and performance analysis, etc.”

– Barry W. Boehm, RAND Corp. 1972

# How long will this be relevant?

- Viewing the adapter in a connected architecture as a first-class citizen applies broadly
  - mobile
  - TV
  - even PCs!

# How long will this be relevant? The Living Room Environment

- A box unifying Internet, Media and a TV display a probably here to stay
  - Today: Cable STB, Blu-ray Player



# How long will this be relevant? The Living Room Environment

- Living room devices are cost-sensitive

Walmart  Christmas costs less at Walmart.

**One Day In-Store Specials** [LEARN MORE ▾](#)

**Saturday Nov. 7th 8:00 am**

**Panasonic** \$788  
Panasonic® Plasma HDTV 46" diagonal screen, TC-P46U1

**HP** \$298  
HP® 250GB PC, 15.6" diagonal display, HPG60-519WM (minimum 10 per store)

**SONY** \$148  
Sony® Blu-Ray Disc™ Player, 508268 (minimum 10 per store)

**XBOX** \$199  
Xbox 360™ Arcade Console with \$100 gift card. (minimum 10 per store)

**MAGNAVOX** \$29  
Magnavox® DVD player with 1080p upconverter, 545538

**SHARP** \$498  
Sharp® LCD HDTV 42" diagonal screen, LC42SB45UT (not available in all stores)

 Print 31069  Find a Store

Quantities of all items are limited. No rainchecks. Prices and availability may vary in AK, HI, OK and WI. Not valid in Puerto Rico or on Walmart.com.

# How long will this be relevant? The Living Room Environment

- Consumer devices pose debugging challenges
  - Standards, and that pesky free market
  - h4k0rz



# How long will this be relevant? The Living Room Environment

- Living room devices are kept a long time
  - 7-15 years for Cable STB, DVD player

If it ain't  
**BROKE**  
*don't fix it!*

# **How long will this be relevant?**

## **The Living Room**

- Today, TV-connected devices are about as fast as home computers were 15-25 years ago.

# How long will this be relevant? The Living Room

- Today, TV-connected devices are about as fast as home computers were 15-25 years ago.

## **BOLD PREDICTION:**

- In 15-25 years, the device connected to the average TV may be as fast as home computers are today.

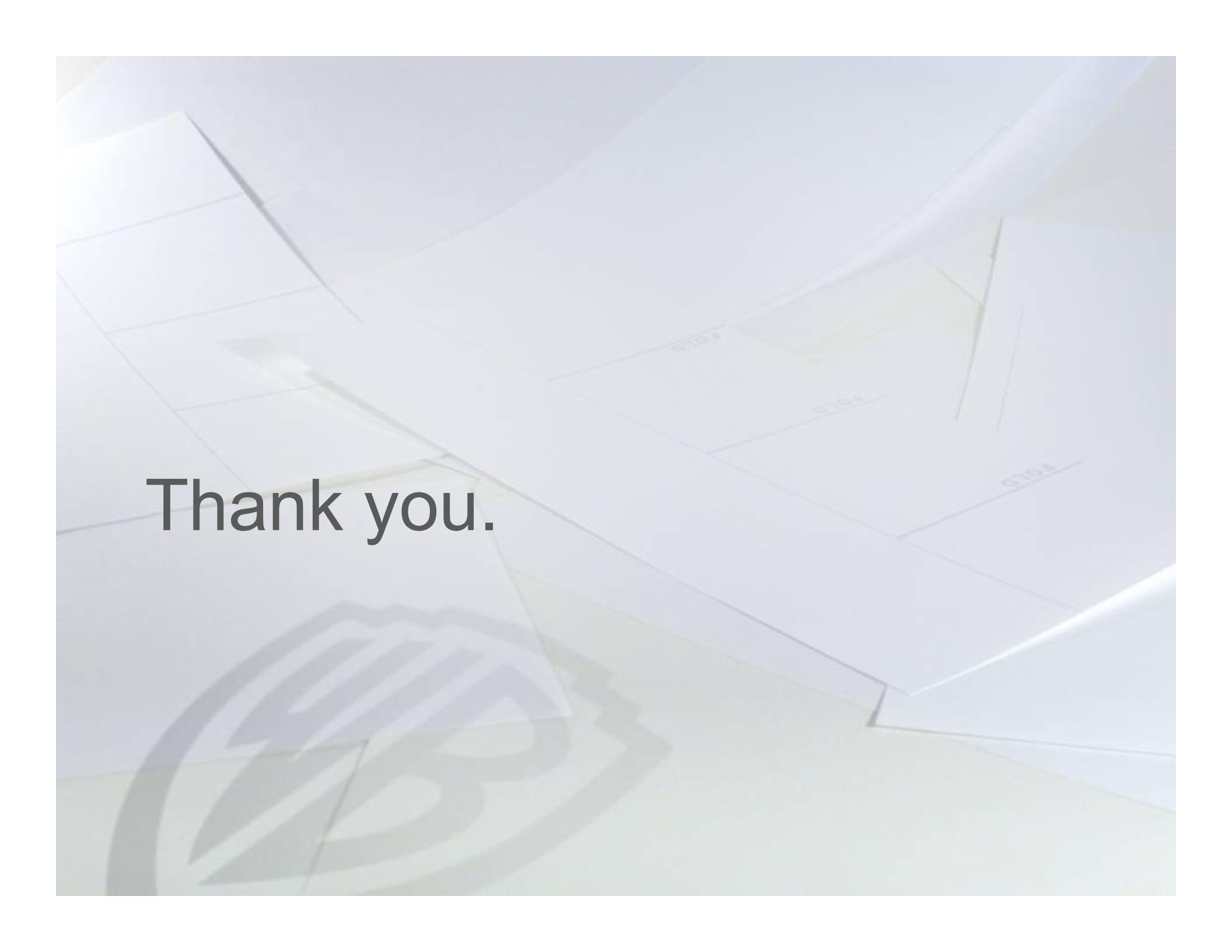
# Concluding Thoughts

## What Shapes Systems?

- Human Behavior
- Market Forces
- Network Performance
- Hardware
- Software
  - High-level Languages
  - Structured Programming
  - OO

# Concluding Thoughts

- In the consumer space, good system architecture is essential.
- Hardware will get faster, but unevenly.
- Systems design lessons outlive particular hardware configurations
  - In 2050, system architecture still won't be network topology!

The background of the image is a light gray color with a subtle, abstract geometric pattern. It features several thin, light-colored lines forming a grid-like structure, with some lines being darker and more prominent. In the lower-left quadrant, there is a large, semi-transparent watermark or logo consisting of stylized, overlapping shapes in shades of gray and white. The text "IBM" is visible within this graphic. Overlaid on the right side of the image is a large, bold, black text that reads "Thank you.".

Thank you.