Music & the Internet
MUMT301

Gabriel Vigliensoni
Schulich School of Music
McGill University

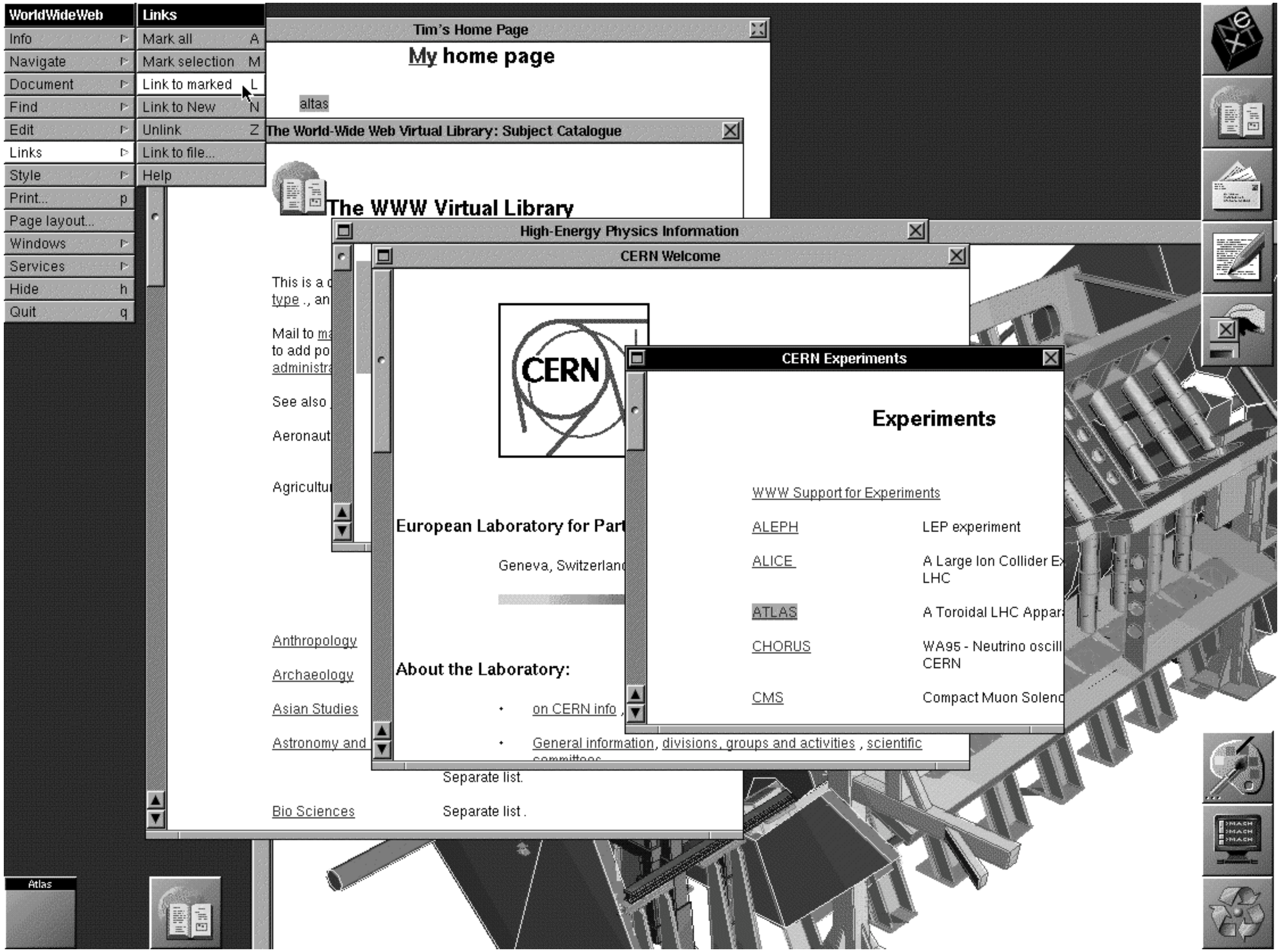
Plan

- Mid-term recap
- Assignment #6 recap
- API and Webservices
- Music APIs
- JavaScript
- Assignment #7

Protocols and expansion

- 1973: Bob Metcalfe@Xerox PARC developed the **Ethernet** technology
- 1976: compact and simple TCP implementation was designed for first **personal computers**
- 1980s: widespread creation of LANs and use of PCs lead to some management issues:
  - There were too many numeric addresses, and so **hosts were assigned names**
  - a **single table** of hosts and names was **no longer feasible**
- 1983 was an important year for the Internet:
  - Paul Mockapetris invented the **Domain Name System** (DNS), allowing to resolve (map) **hierarchical host names** into an Internet address
  - **Internet protocols** were incorporated natively into the **Unix OS** at UC Berkeley, which led to a widespread adoption of the Internet into the research community
  - ARPANET moved from its own host protocol to TCP/IP

History of WWW: Browsers



Tim's Home Page

My home page

altas

The World-Wide Web Virtual Library: Subject Catalogue

The WWW Virtual Library

This is a d
type ., an

Mail to ma
to add po
administra

See also

Aeronaut

Agricultu

Anthropology

Archaeology

Asian Studies

Astronomy and

Separate list.

Bio Sciences Separate list .

High-Energy Physics Information

CERN Welcome

CERN

European Laboratory for Part

Geneva, Switzerlan

About the Laboratory:

- on CERN info ,
- General information, divisions, groups and activities , scientific committees

CERN Experiments

Experiments

WWW Support for Experiments

ALEPH	LEP experiment
ALICE	A Large Ion Collider E LHC
ATLAS	A Toroidal LHC Appar
CHORUS	WA95 - Neutrino oscill CERN
CMS	Compact Muon Solen

1990: WorldWideWeb by Tim Berners-Lee

Structured Audio

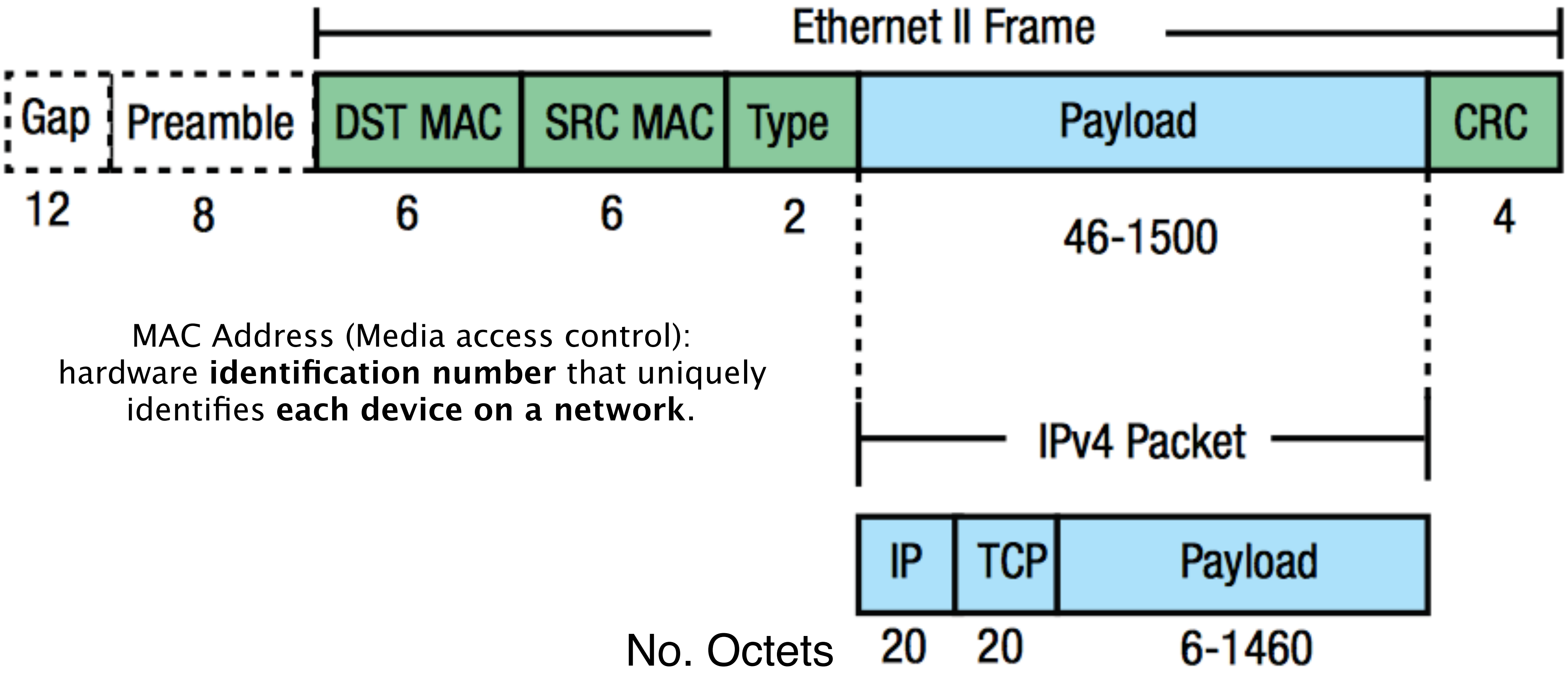
- Structured audio formats provide **data to support dynamic construction of sound** through hardware and software
- *Sequencers* and *trackers* control
  - the **timing of sounds**, i.e., when individual sound elements start or stop
  - **sound attributes** such as volume, pitch, and other features
- *Sound elements* can be
  - short sections of **sound samples or loops** or
  - **data elements that characterize a sound** so that a *synthesizer* can produce the actual sound

Internet stats

- Internet usage and population in the world
  - http://www.internetworldstats.com/stats.htm
- Internet traffic
  - http://www.internetlivestats.com/

Complete Ethernet Packet

Taken from openmicrolab.com



No. Octets

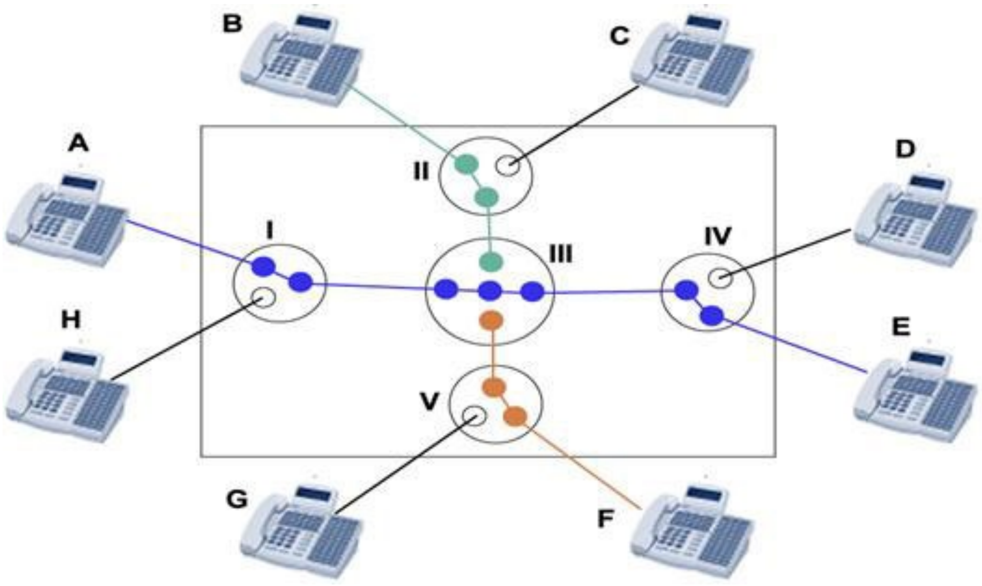
MAC Address (Media access control):
hardware **identification number** that uniquely
identifies **each device on a network**.

Compressed sound file formats

- Lossy and Lossless
- **Lossy compression**:
  - only **an approximation of the original data can be reconstructed after decompression**,
  - How well it approximates the original data **depends on the compression rate**
  - Common lossy formats
    - MP3 (patented!)
    - Vorbis (aka Ogg Vorbis), xiph.org webpage (free and open source) (V1.0 2002)
    - WMA
- **Lossless compression**:
  - **data can be perfectly reconstructed after decompression**
  - Lossless formats
    - Monkey's Audio, WavPack, Apple Lossless, ...
    - FLAC, xiph.org webpage
      - non-proprietary
      - no patent restricted
      - open-source

Switching

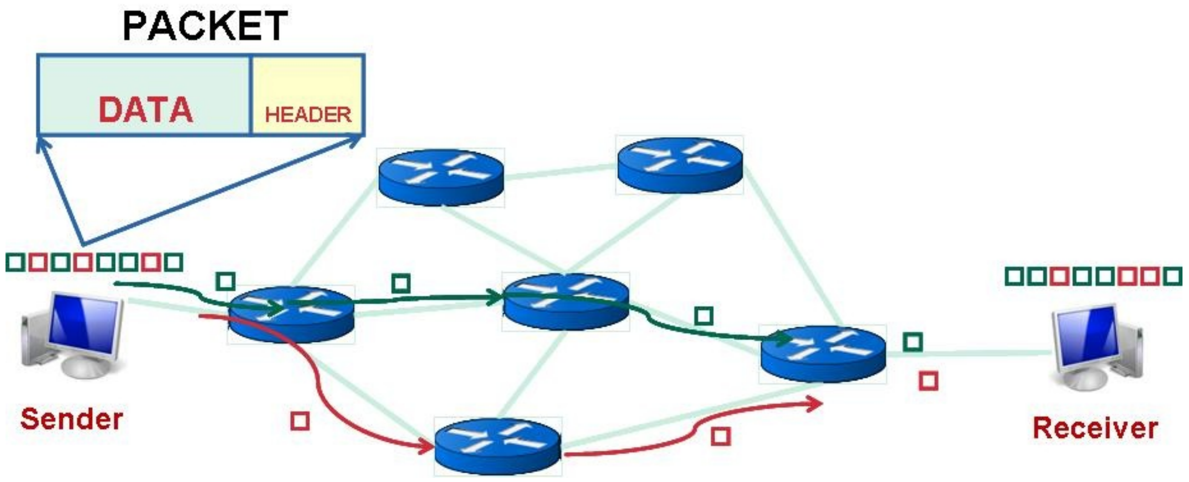
- Switching is the method by which data is transferred from an input port to an output port
- In **circuit switching**:
  - a path is first **reserved**
  - data is transferred after the **connection has been established**
  - all data passes through the **same circuit**
  - **no other user can use** the circuit until the session is completed
  - the **circuit is released afte**r the data is transferred



Taken from http://cyberlawsolutions.blogspot.ca/2011/12/packet-switching.html

Packet switching

- In **packet switching**
    - Divides the data to be transmitted into **small units** (packets) **transmitted independently** through the network
    - Each packet may be **routed via a different path**
    - The **original message is reassembled** in the correct order at the destination based on the packet number
- A **packet consists of**:
    - **Source**: the IP address of the computer sending the packet
    - **Destination**: the IP address of the destination computer
    - **Length**: the length of the packet in bytes
    - **Number**: the total number of packets in the complete message
    - **Sequence**: the number of this packet in the whole list of packets making up this communication



Taken from http://computernetworkingsimplified.com/physical-layer/overview-circuit-switching-packet-switching/

The musician's POV

- None of the digital music services will let you upload your music directly
- Artists need to be signed to a major (or big indie) record label
- Or use a **digital music distributor** to get in a digital music service
  - Tunecore
  - CDBaby
  - Mondotunes
  - JTV Digital
  - Zimbalam
  - ReverbNation
  - DittoMusic
  - Routenote
  - Distrokid
  - Soundrop
- Digital music distribution across the globe

Widespread infrastructure

- **US Federal agencies** made and implemented **policy decisions** that shaped Internet by the late 80s
  - Federal agencies **shared the cost of common infrastructure** (e.g., trans-oceanic circuits)
  - NSF encouraged regional networks to **look for non-academic customers** to lower costs
  - However, NSF **prohibited the use of its national network for non-academic or research purposes** with the intention of **stimulate the growth of private networks**
  - 1995: The NSF national network was defunded in 1995, but its policies led the Internet to grow to around 30,000 networks just in the US
- 1995: FNC passed a resolution defining the term Internet:
  - **"Internet" refers to the global information system that:**
    - is linked together by a globally **unique address space** based on the IP
    - is able to **support communications using TCP/IP**
    - provides **high level services** layered on the communications and infrastructure previously described

API and Webservices

- **API** (Application Programming Interface)
    - Specifies a software component in terms of:
        - their **inputs** and **outputs**
        - the underlying **types**
        - its **operations**
    - APIs can come as a **specification** of **remote calls** exposed to the API consumers

API and Webservices

- Web service:
  - "...a software system designed to support interoperable **machine-to-machine interaction** over a network." (W3C)
  - **method of data exchange** that **doesn't depend upon a particular programming language**
  - Web services can be **used by other applications**

REST

- REpresentational State Transfer
- Set of principles for creating web services
- Language and platform independent
- Uses HTTP or HTTPS
- HTTP-based RESTful APIs are defined with these aspects:
  - base URI, such as `http://example.com/resources/`
  - standard HTTP methods (e.g., GET, PUT, POST, or DELETE)
  - an Internet media type for the data. This is often XML or JSON but can be any other valid Internet media type
- Message format can be: XML, JSON, HTML, plain text, etc.

Message formats JSON/XML

- JSON (JavaScript Object Notation)
    - open-standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs
    - http://json.org/example.html
- Download a JSON viewer for your browser
    - http://jsonview.com/example.json
- XML and JSON example
    - http://musicbrainz.org/ws/2/artist?query=ratatat&fmt=xml
    - http://musicbrainz.org/ws/2/artist?query=ratatat&fmt=json

API and Webservices

- Examples:
  - Weather
  - Exchange rate
  - Stock prices
  - Social data: Instagram, Twitter, Facebook
  - Music APIs!

Music APIs

- MusicBrainz API
- LastFM API
- Echonest API (RIP)
- and many others ...
    - http://musicmachinery.com/music-apis/

MusicBrainz API

- An **interface** to the MusicBrainz Database
  - https://musicbrainz.org/doc/MusicBrainz_Database/Schema
- Aimed at any applications requiring **music metadata**
- The service's architecture follows the **REST design** principles
- Interaction with the web service is done **using HTTP** and all **content is served in XML and JSON**
- https://musicbrainz.org/doc/Development
  - The **web service root URL** is http://musicbrainz.org/ws/2/
- **Search:**
  - http://musicbrainz.org/ws/2/artist?query=ratatat
- **Lookup:**
  - http://musicbrainz.org/ws/2/artist/f467181e-d5e0-4285-b47e-e853dcc89ee7
- **Query:**
  - http://musicbrainz.org/ws/2/release?artist=f467181e-d5e0-4285-b47e-e853dcc89ee7&type=album
  - http://musicbrainz.org/ws/2/release-group?artist=f467181e-d5e0-4285-b47e-e853dcc89ee7&type=album

LastFM API

- An interface to the LastFM Database
- Aimed at developing musical applications requiring music metadata and listener's listening behaviour data
- The service's architecture follows the REST design principles
- Interaction with the web service is done using HTTP and all content is served in XML format
- The same web service is also available in JSON format
- https://www.last.fm/api

Echonest API

- An interface to various Echonest APIs
- Aimed at developing musical applications requiring music metadata, cultural metadata, and acoustic features data
- The service's architecture follows the REST design principles
- Interaction with the web service is done using HTTP and all content is served in XML format
- The same web service is also available in JSON format
- http://developer.echonest.com/docs/v4
- https://echonest.github.io/remix/
- Examples:
  - http://static.echonest.com/labs/
- Spotify API

BREAK

Potential final projects

- Music recommendation site
- Web-based musical instrument or controller
- Music playlist maker
- History of the recording industry in the age of the Internet
- Statistical / historical analysis of music industry based on web-based data
- Study of international music copyright laws
- Study of fair use and copyright infringement music cases
- Substantial music composition (20-30 min) strictly using web resources with substantial write up (2-3 pages)
- Comprehensive comparison of on-demand music streaming services

JavaScript

- mumt301.github.io
- In-class assignment: create a dynamic artist page with picture and bio
- Assignment #7