Beyond REST

Yogeshwar Srikrishnan
API Evolution

- EDI
- CORBA
- DCOM
- Java RMI
- XML RPC
- SOAP
REST

- Client-Server: Separation of concerns
- Client/Stateless-Server: Visibility, scalability
- Caching: Performance improvements
- Uniform Interface: Consistent interactions and better visibility of interactions. Ease of adoption.
- Layered System: Simplified components and improved scalability
- Code-On-Demand: Allows extensibility.
REST vs. SOAP: Simplicity wins again

Distribution of API protocols and styles

Based on directory of 3,200 web APIs listed at ProgrammableWeb, May 2011
A **presumptive architecture** is a software architecture (or a family of architectures) that is dominant in a particular domain. Rather than justifying their choice to use it, developers in that domain may have to justify a choice that differs from the presumptive architecture.

George Fairbanks
"Those who cannot remember the past are condemned to repeat it,"

Philosopher “George Santayana”
REST - Challenges

- OSFA
- Multiple Round Trips
- Versioning
- Style Versus Standard
- Primitive Tools

- REST endpoints are usually weakly-typed and lack machine-readable metadata
- Additional Use Cases
Orchestration/Experience API

- LSUD versus SSKD
- Device-specific Wrappers
- Query-based APIs
- Experience Based APIs
GraphQL

- GraphQL Query
- Hierarchical
- Strongly typed
- Introspective
- Version Free
- powers hundreds of billions of API calls a day
user(id: 4802170) {
    id
    name
    isViewerFriend
    profilePicture(size: 50) {
        uri
        width
        height
    }
    friendConnection(first: 5) {
        totalCount
        friends {
            id
            name
        }
    }
}

data: {
    user: {
        id: "4802170",
        name: "Lee Byron",
        isViewerFriend: true,
        profilePicture: {
            uri: "cdn://pic/4802170/50",
            width: 50,
            height: 50
        }
    },
    friendConnection: {
        totalCount: 13,
        friends: [
            {
                id: "305249",
                name: "Stephen Schwink"
            },
            {
                id: "3108935",
                name: "Nathaniel Roman"
            }
        ]
    }
}
FALCOR

- A JavaScript library for efficient data fetching.
- Data is the API
- Falcor Model/ Asynchronous API.
- Caching
- Batching
- Request Deduping
Binary Protocols

- Efficient
- Speed of Transmission and Interpretation.
- Native bindings
- Ease and Speed Of Development
- Better Versioning Support
- IDL – Interface Description Language
- Non – Browser/UI Consumption
- Less Boilerplate Code
Protocol buffers are a language-neutral, platform-neutral extensible mechanism for serializing structured data.

Thrift is a software library and set of code-generation tools developed at Facebook to expedite development and implementation of efficient and scalable backend services.

Apache Avro™ is a data serialization system.

---

<table>
<thead>
<tr>
<th>Protocol Buffers</th>
<th>Apache Thrift™</th>
<th>Apache Avro™</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google 2008 BSD License</td>
<td>Facebook 2007 Apache License IDL Cleaner RPC Entire integrated RPC framework and many choices</td>
<td>Data Stored along with Schema Apache JSON Format RPC</td>
</tr>
</tbody>
</table>

- Google 2008 BSD License: stable Well tested.
- gRPC
- facebook
- Pig load/store function
- Apache HBase gateway server
- Active MQ
- Evernote
- Apache HBase gateway server
Asynchronous API

- Loosely Coupled
- Well Distributed
- Event-driven architecture (EDA)
- IOT – Internet of Things
- MOM/Message Broker
AMQP

- AMQP 1.0 is an international standard, approved by ISO and IEC
- To become the standard protocol for interoperability between all messaging middleware
- AMQP is an efficient, reliable, wire-level messaging protocol
- Apache Qpid, RabbitMQ, Windows Azure Service Bus
Apache KAFKA

- Apache Kafka is publish-subscribe messaging rethought as a distributed commit log.
- Simple, high-performance, language agnostic TCP protocol.
- Fast
- Scalable
- Durable
- Distributed by Design
REST Standards

- Internal Standards
- http://www.odata.org/getting-started/ OASIS
- Open API - Swagger
Prepare For Change

- Build awareness around new set of Technologies.
- Be early adopters.
- Standards and Choices
References

http://www.infoq.com/articles/rest-soap
https://netflix.github.io/falcor/starter/what-is-falcor.html
https://www.amqp.org/
http://blog.cloudera.com/blog/2014/09/apache-kafka-for-beginners/
Thank you