KubeCon | CloudNativeCon

North America 2017

REST, RPC, and Brokered Messaging

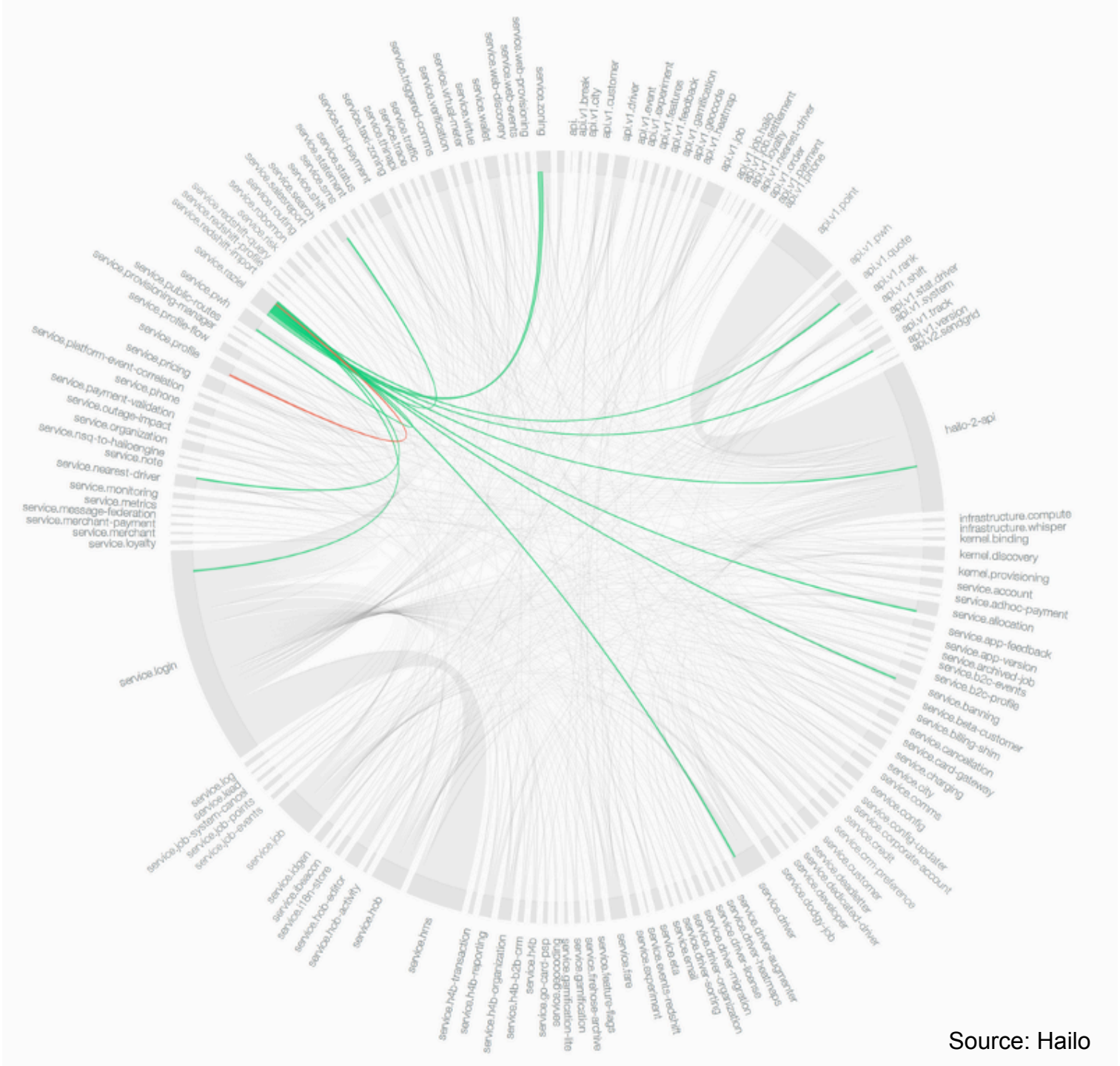
Nathan Murthy, Staff Software Engineer, *Tesla*

TESLA

Automotive

Energy


TESLA



Source: Hailo

UNIVERSITY OF CALIFORNIA, IRVINE

Architectural Styles and the Design of Network-based Software Architectures

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Roy Thomas Fielding

2000

Implementing Remote Procedure Calls

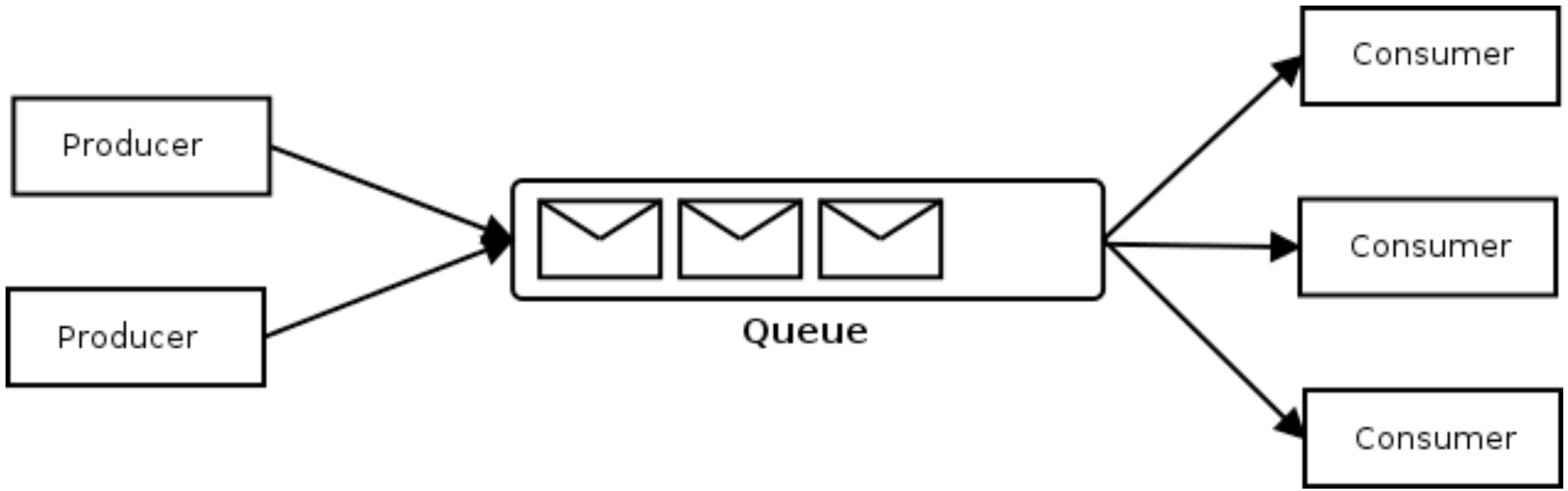
ANDREW D. BIRRELL and BRUCE JAY NELSON
Xerox Palo Alto Research Center

Remote procedure calls (RPC) appear to be a useful paradigm for providing communication across a network between programs written in a high-level language. This paper describes a package providing a remote procedure call facility, the options that face the designer of such a package, and the decisions we made. We describe the overall structure of our RPC mechanism, our facilities for binding RPC clients, the transport level communication protocol, and some performance measurements. We include descriptions of some optimizations used to achieve high performance and to minimize the load on server machines that have many clients.

CR Categories and Subject Descriptors: C.2.2 **[Computer-Communication Networks]:** Network Protocols—*protocol architecture*; C.2.4 **[Computer-Communication Networks]:** Distributed Systems—*distributed applications, network operating systems*; D.4.4 **[Operating Systems]:** Communications Management—*message sending, network communication;* D.4.7**[Operating Systems]:** Organization and Design—*distributed systems*

General Terms: Design, Experimentation, Performance, Security

Additional Keywords and Phrases: Remote procedure calls, transport layer protocols, distributed naming and binding, inter-process communication, performance of communication protocols.


Producer
Producer
Queue
Consumer
Consumer
Consumer

When REST makes sense

- Layered hypermedia data transfer (HTTP)
- Ideal for web browsers and front-end apps
- Client-Server communication
  - User Agents
  - One-to-One
- Human-readable content
- Enforcing CRUD semantics
  - POST, GET, PUT/PATCH, DELETE
- Creating public or external APIs for web clients

When RPC makes sense

- Protocol agnostic, doesn't *have* to be HTTP but it could
- Inter-service communication
  - Services to talk to other services
  - One-to-One
- Human is not necessarily producer/consumer of data
- Free-form semantics, not limited to CRUD
  - You get to pick your own nouns and verbs
  - Affords stronger domain specificity
- Creating internal APIs inside the data center

What about Message Brokers?

- Many-to-Many
  - Abstracts away locations of producers and consumers
  - Pub-Sub
- Asynchronous communication
  - Message Queuing
  - Decouple requests and responses
- Long-lived connections
- Moving large volumes of data in streams
- Command Query Responsibility Segregation (CQRS)
- Event Sourcing

Square Peg, Round Hole

- Avoid making an awkward domain language
- Objects at REST tend to stay at REST
  - Message Broker topics mimicking HTTP verbs
    - /control/**post**/device
    - /measurement/**get**/device
- Different styles of “real-time” HTTP/1.1
  - Long polling
  - WebSockets, Comet
  - `Content-Type: text/event-stream`
  - `Transfer-Encoding: chunked`
- Just use HTTP/2 streaming

Putting Things Together

CLOUD NATIVE
COMPUTING FOUNDATION

Tools for REST and RPC

linkerd

OpenTracing

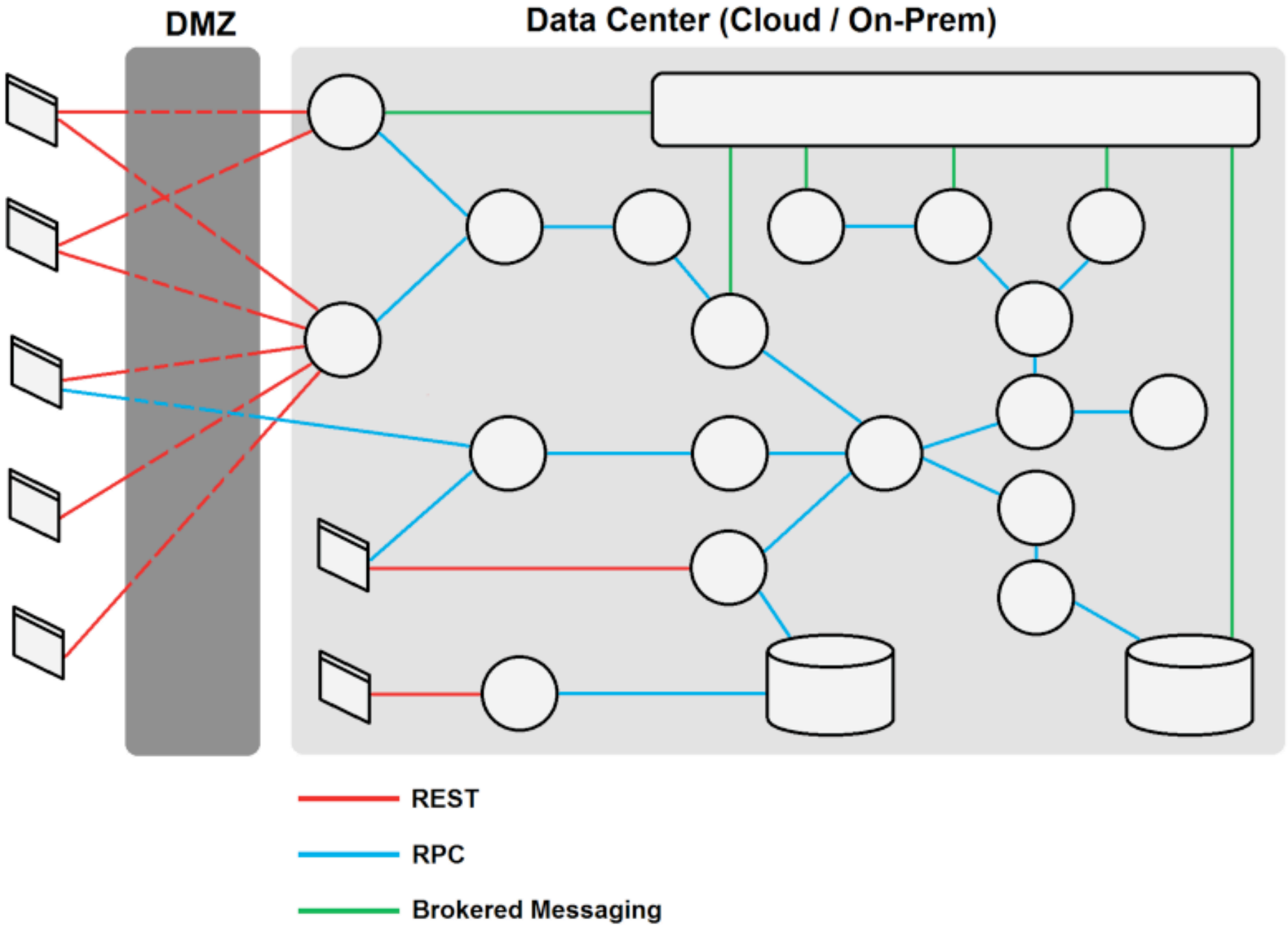
gRPC

Envoy

Jaeger

Tools for Brokered Messaging

???


DMZ
Data Center (Cloud / On-Prem)
REST
RPC
Brokered Messaging

Thank You!

@natemurthy