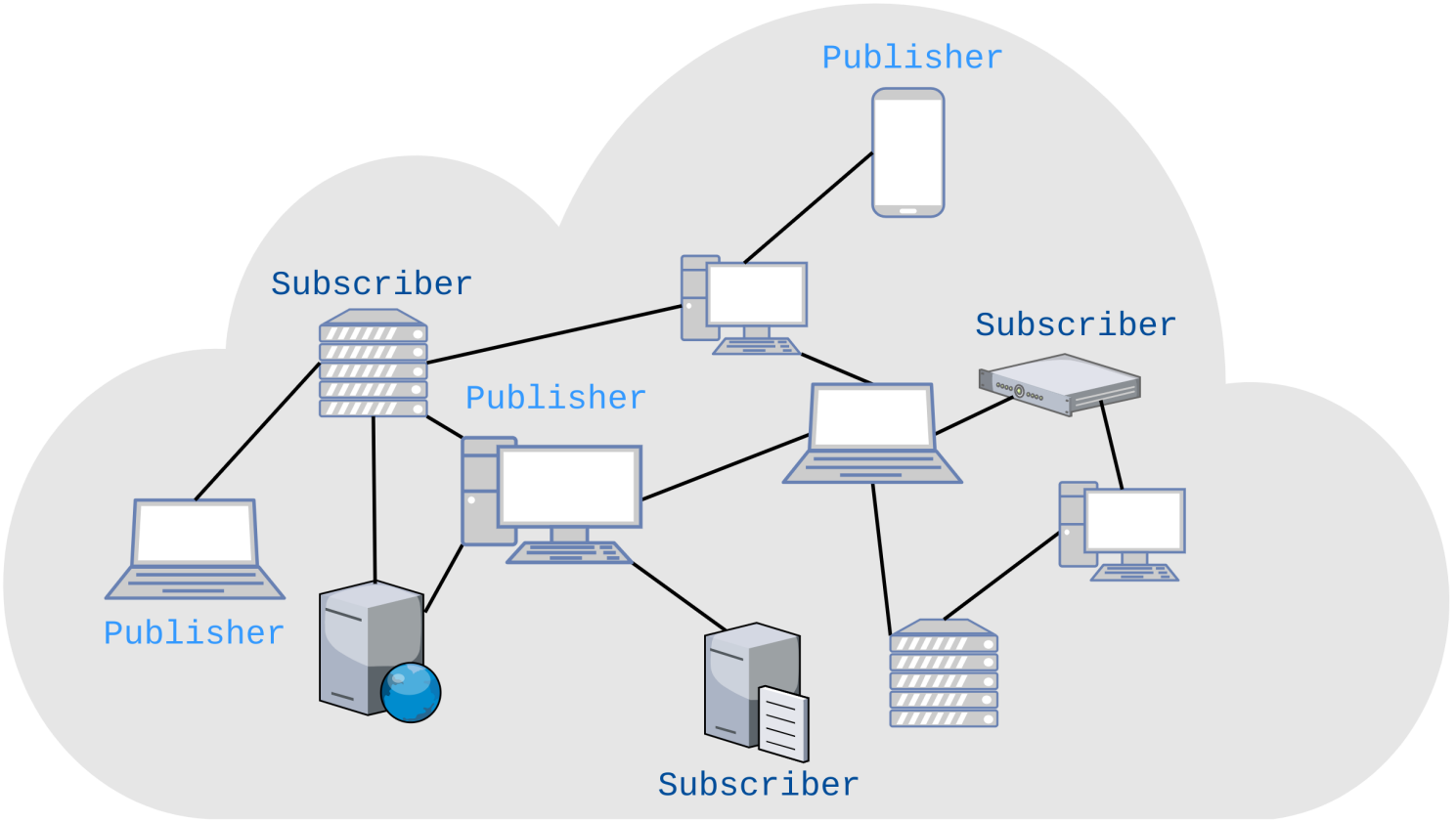
PulsarCast

Scaling Pub-Sub over the distributed web

João Gonçalo da Silva Antunes
Instituto Superior Técnico
INESC-ID

Motivation

World Wide Web



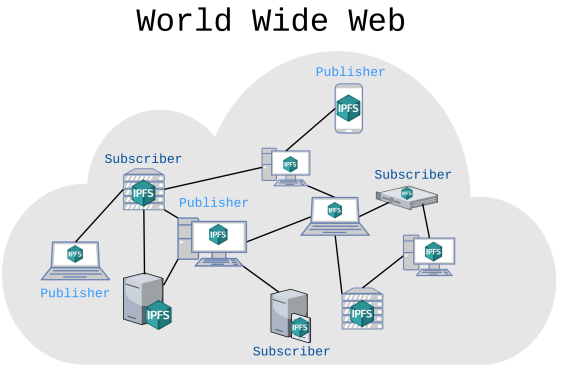
Problems

Lack of a pub-sub system that:

- Scales for a really large number of participants (web)
- Provides reliability:
  - Delivery guarantees
  - Data persistence

Objectives

- Decentralised pub-sub system using IPFS[1]:
  - Highly scalable
  - Reliable
  - Assures persistence

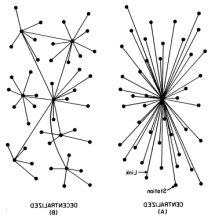


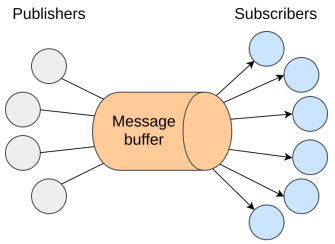
- Develop a solution that meets these requirements
- Evaluate its performance on multiple environments

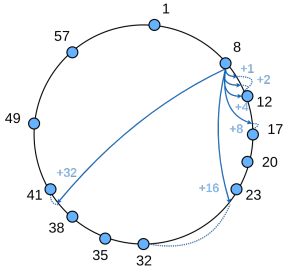
[1] - https://ipfs.io/

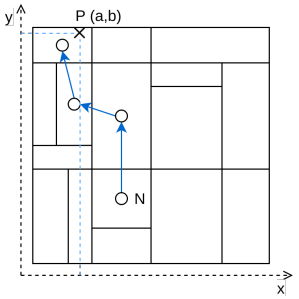
Related Work

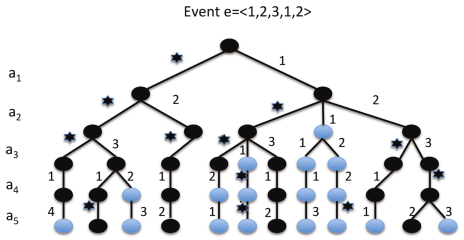
Pub-Sub Systems



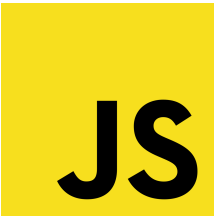








Web Technologies



Pub-Sub Paradigm

- Communication paradigm providing full decoupling in:
  - Time
  - Space
  - Synchronisation

Design Dimensions

- Subscription model
- Network architecture
- Overlay structure
- Subscription management
- Event dissemination

Subscription Model

- Topic based
  - E.g. Redis[2], Scribe, Tera, Poldercast
- Type based
  - E.g. Hermes
- Content based
  - E.g. Gryphon, Siena, Meghdoot

[2] - https://redis.io/

Topic Based

- ✔ A simple notion of group
- ✔ Members of the group receive every message
- ✔ Can build a complex hierarchy
- ✗ Lacks expressiveness

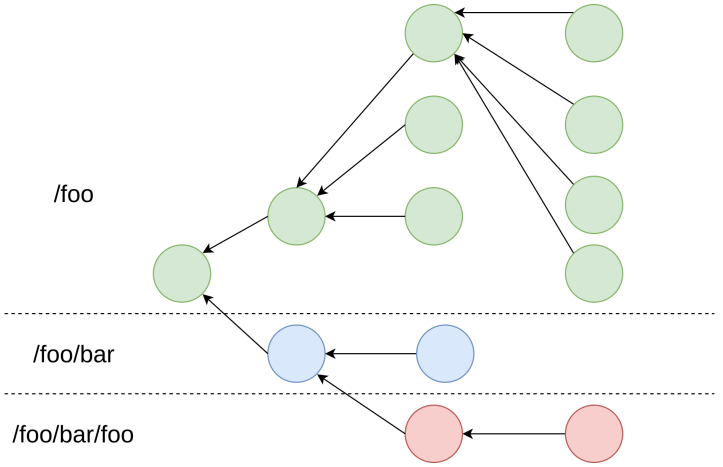


Fig.1: Events on a topic hierarchy

Content Based

- ✔ Filter events based on their content
- ✔ Support for expressive subscriptions
- ✗ Requires complex filtering and message forwarding

Message

```
{
  exchange: "Euronext Lisboa",
  company: "CTT",
  order: "buy",
  number: "100",
  price: "5.55",
}
```

Subscription

```
{
  exchange: "Euronext Lisboa",
  order: "buy",
  number: ">50",
  price: "<10",
}
```

Fig.2: JSON subscription example

Network Architecture

- Centralised:
  - E.g. Redis, Kafka[3], Gryphon
- Decentralised:
  - E.g. Scribe, Meghdoot, Poldercast

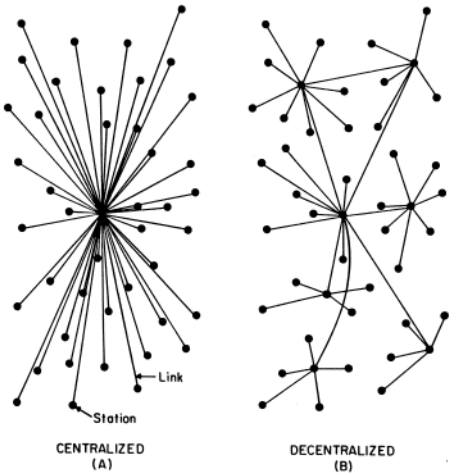


Fig.3: Network architecture overview

[3] - http://kafka.apache.org/documentation/#introduction

Centralised

- Focus:
  - ✔ Reliability
  - ✔ Consistency
- Lacks:
  - ✗ Scalability
  - ✗ Data Throughput

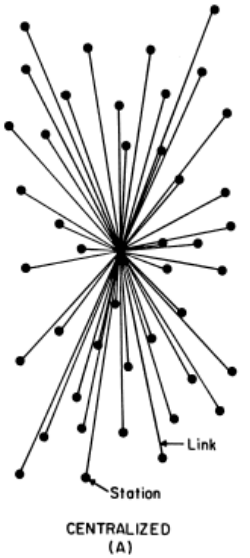


Fig.4: Centralised architecture example

Decentralised

- Focus:
  - ✔ Scalability
  - ✔ Data Throughput
- Lacks:
  - ✗ Reliability
  - ✗ Consistency
  - ✗ Persistence

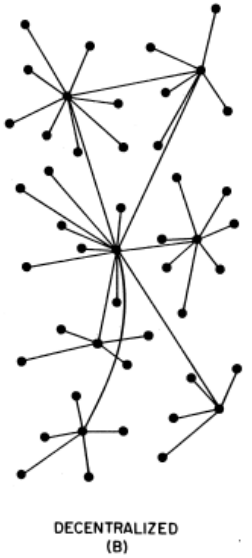
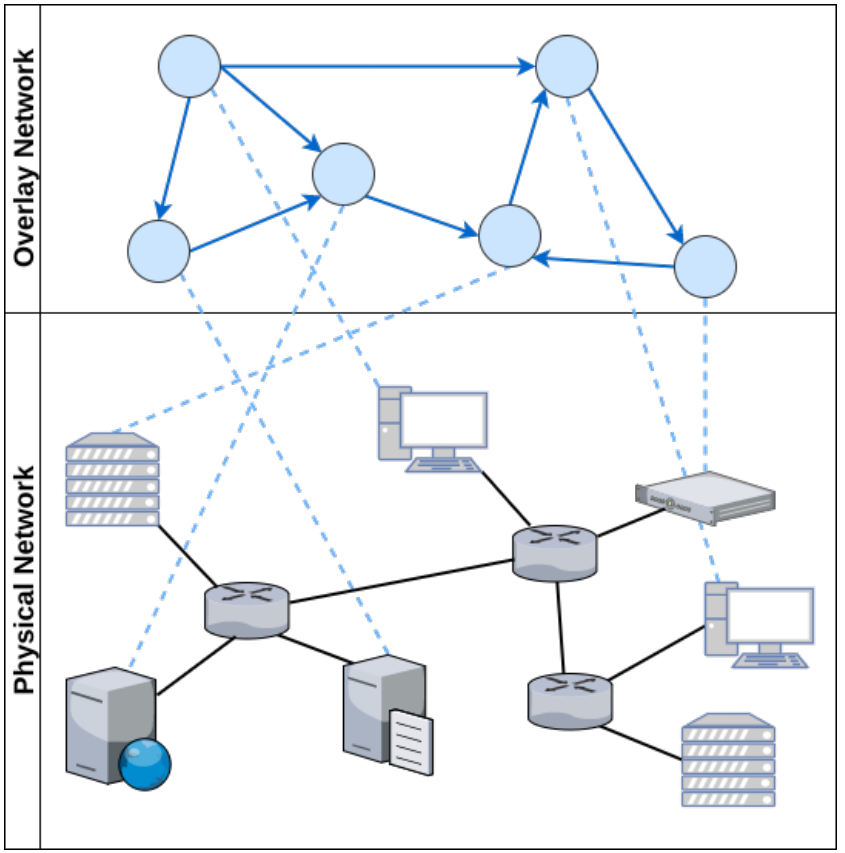


Fig.5: Decentralised architecture example

Overlay Structure



Unstructured Overlay

- Each peer connects to a subset of nodes
- No clear structure or hierarchy
- Usage of gossip based membership protocols (e.g. Cyclon[6]) to help preserve:
  - Network diameter
  - Average degree

Structured Overlay

- Peers organise according to a specific structure (e.g. ring, tree, multi-dimensional space).
- Common approach is to use a Distributed Hash Table
  - E.g. Chord, Kademlia, CAN

DHT

- Peer identifiers evenly spread across key space
- Ensures the content is evenly distributed
- Queries usually solved in logarithmic time

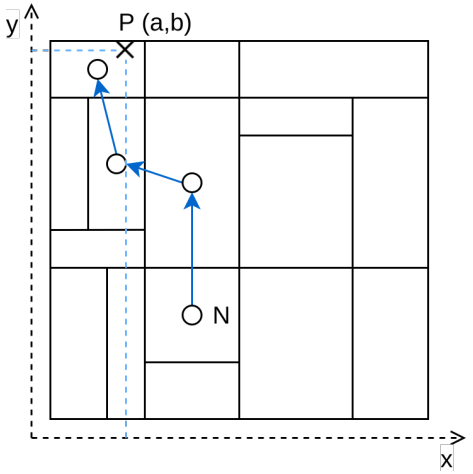


Fig.6: 2 dimensional CAN routing

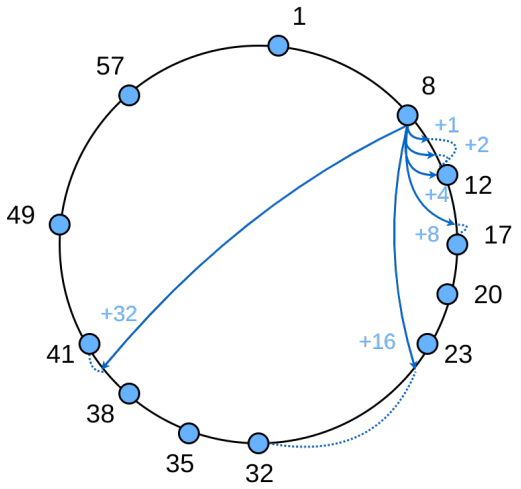


Fig.7: Chord ring

Subscription Management

Event Dissemination

System Analysis

	Subscription Model	Architecture	Overlay Structure	Subscription Management	Event Dissemination
Gryphon	Content	(C) broker hierarchy	N/A	Each broker responsible for a subscription scheme	Tree hierarchy
Siena	Content	(C) broker mesh	N/A	Keep state at each node for subscription routes	Flood with cached state
Jedi	Content	(C) broker hierarchy	N/A	Keep state at each node for subscription routes	Tree hierarchy
Bayeux	Topic	Decentralised	Tapestry DHT	Rendezvous node	Multicast tree
Scribe	Topic	Decentralised	Pastry DHT	Rendezvous node	Multicast tree
Meghdoot	Content	Decentralised	CAN DHT	Point in CAN DHT	CAN routing
Hermes	Type	Decentralised	Pastry DHT	Rendezvous node	Multicast tree
Tera	Topic	Decentralised	Gossip based overlay	Unstructured overlay per topic	Random walks & flooding
Mercury	Content	Decentralised	Ring based DHTs	Overlay per attribute in schema	Route through ring overlays
Sub-2-Sub	Content	Decentralised	Ring based DHT & gossip overlay	Clustering of similar subscriptions	Gossip & ring overlay routing
Poldercast	Topic	Decentralised	Ring based DHT, Vicinity & Cyclon	Ring overlay per topic	Ring overlay routing

(C) – Centralised

System Analysis

	Relay Free Routing	Delivery Guarantees	Fault Tolerance
Gryphon	N/A	yes	Best effort
Siena	N/A	yes	Best effort
Jedi	N/A	yes	Best effort
Bayeux	No	yes	Best effort, no subscription persistence
Scribe	No	yes	Best effort, no subscription persistence
Meghdoot	No	yes	Replicated subscriptions
Hermes	No	yes	Best effort
Tera	No	no	Best effort
Mercury	No	yes	Best effort
Sub-2-Sub	No	no	Best effort
Poldercast	Yes	Yes (every publisher is a subscriber)	High resilience to churn, no subscription persistence

Delivery Guarantees – Event delivery guarantees under normal network conditions
Fault Tolerance – Mechanisms to guarantee successful event delivery and subscription matching under heavy churn

Web Technologies

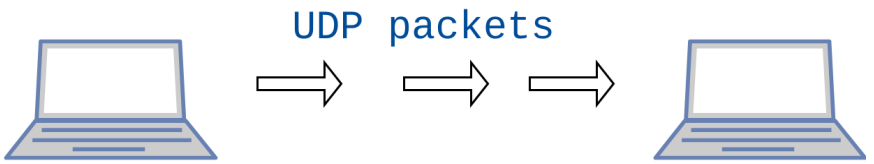
- The Javascript ecosystem
- New network protocols that facilitate P2P communication
- New P2P applications that leverage all of this
  - E.g. IPFS

The Javascript Ecosystem

- Javascript is one of the main programming languages for the web
- Thanks to NodeJS, now runs in servers also
- Its package manager (NPM), is one of the largest package registries in the world
- NPM powers a UNIX-like culture of small modules that work well together

New Protocols

- WebRTC made possible full-duplex communication between browsers without an intermediary.
- QUIC and uTP provided alternatives to TCP, bringing reliability and order delivery over UDP



IPFS

A **P2P hypermedia** protocol designed to create a **persistent**, **content-addressable** network for the web

-

Proposed Solution

Overlay Structure

Data Structures

Subscription Management

Event Dissemination

- (mention matching)

Quality of Service

Evaluation

Conclusion

Q&A

References