On the Creation & Discovery of Topics in Distributed Publish/Subscribe systems

Shrideep Pallickara, Geoffrey Fox & Harshawardhan Gadgil

Community Grids Lab, Indiana University

Messaging Systems

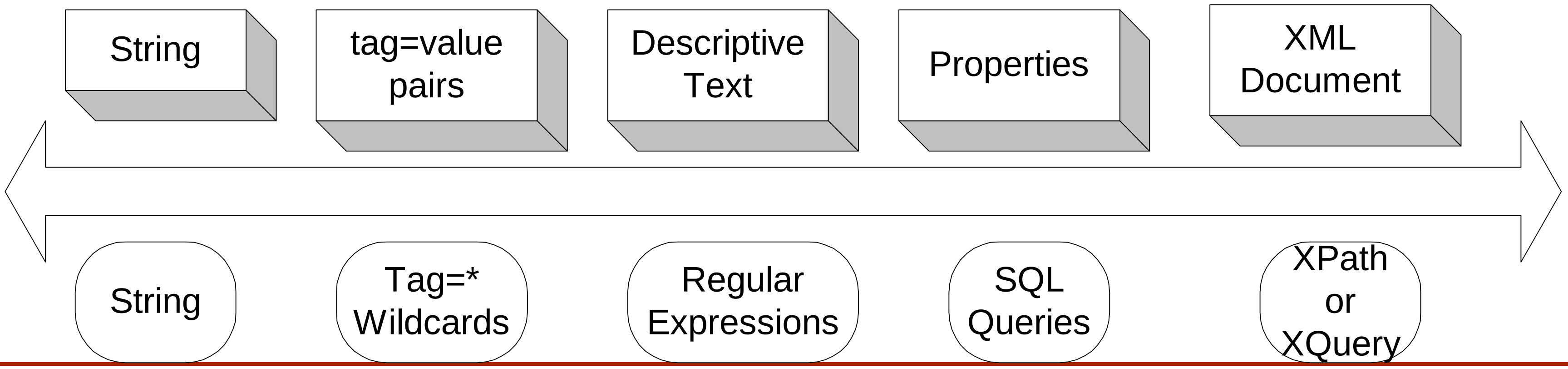
- Messaging is the routing of content from the producer to the consumer.
- This can be point-to-point (involving a single producer and consumer) or many-to-many (involving many producers and consumers).
- Approaches to messaging include systems such as queuing, P2P systems and publish/subscribe.

Publish/Subscribe Systems

- Software multicast
- Routing is based on the message content
- Routing of messages, from the publisher to the subscriber, is within the purview of the middleware
- Gained a lot of traction in recent years.
  - JMS, WS-Notification and WS-Eventing.

Topics and Subscriptions

- A message comprises a set of headers and the payload
- A Topic is a *content descriptor* and is present is all messages.
  - Complexity of a topic varies proportionally with the richness of the content descriptor.
- Subscriptions are constraints specified on these content descriptors (or Topics).
- Depending on the *type* of topics, specified subscriptions vary.



Topic related issues

- Topics tend to be treated as communal resources
  - No dissemination constraints; launching attacks is easy.
- No one *owns* a Topic, so policies cannot be enforced.
- No discovery of topics.
  - Topics to publish or subscribe to are established in an out-of-band fashion (typically hard-coded).
- No concept of lifecycle management for Topics.
  - Once created, topics are alive forever: no garbage collection
  - In some systems lifecycles associated with subscriptions
- Collisions in the topic space
- Problems increase as the number of topics increase

Features of our framework

- Scheme for creation and advertisement of Topics
  - Establish provenance: Precursor to enforcing policies
  - Establish lifetimes for topic: Garbage collection of topics.
  - Topics are guaranteed to be unique across the system
- Facilitates discovery of topics.
  - Mandate possession of credentials for discovery.
- Subscribers can subscribe to trusted sources.
- Scheme is asynchronous and resilient to failures.
- Secure creation, advertisement & discovery of topics

Topic Discovery Nodes (TDN)

- Specialized nodes that serve as a repository of topics
- There can be several TDNs within the system
  - Need not be exact replicas of each other
  - A domain may have its own private TDN
- Responsible for the generation of unique topics.
- Establish topic ownership
- Subscribes to the following topics
  - Services/Discovery/Topics
  - Services/Discovery/TopicDiscoveryNode
  - Services/Discovery/TopicDiscoveryNode/TDN-ID

Anatomy of a Topic creation Request

- Creator's certificate including name and institution
- Information about topic type and lifecycle: start & end
- Topic template – TDN adds information to this to make topic unique throughout the system
- Descriptive info to enable discovery of the topic
  - Could be based on Strings, verbose text or XML
  - Discovery queries are evaluated against this part.
- Restrictions on who can discover this topic
- Sign this request to demonstrate private-key possession

Locating a TDN

- Issue a TDN discovery request to a specialized *private topic* or Services/Discovery/TopicDiscoveryNode
- This request contains
  - The requestor's credentials
  - The topic on which responses should be sent back
- A TDN responds based on the presented credentials
  - Also includes the dedicated topic for communications
- There could be one or more responses to the request.
- Requestor chooses TDN based on response times or credentials.

Processing a Topic Creation Request

- The TDN generates a new UUID.
- This UUID is added to the topic template to generate a unique topic.
- UUID generation at TDN prevents a malicious user from claiming someone else's topic as theirs.
- TDN then signs the info supplied in the topic creation request, and the generated topic structure.
  - This is the Topic Advertisement.
- Topic creator posts Advertisement on different TDNs.

Topic Discovery

- Topic Discovery requests are targeted to all *willing* TDNs or a specific TDN (possibly private)
  - Queries can also include start/end times
- At a TDN, the discovery query is evaluated against the descriptions to locate matching topics.
  - Discovery constraints imposed by owner are enforced here.
- Matching advertisements are routed back to requestor.
- Requestor decides on topic based on the advertisement
  - Owner, Institution etc.

Security & Fault Tolerant Aspects

- Topic Creation & Discovery is restricted to possession of valid credentials.
- Once a TDN has been discovered, all exchanges between a TDN and entity are secured.
  - Messages are encrypted with a secret key, the secret key is encrypted with the public-key of the intended recepient.
- TDNs may fail at any time.
  - Topic creation requires only one TDN to be available
  - Discovery requests can be flushed through system, and clients may service these requests.

Performance

	Broker in Tallahassee, TDN at Indianapolis and Client in Bloomington. All results in Milliseconds.				
	Mean	**Std. Dev.**	**Max**	**Min**	**Std. Err**
Topic Creation	641.06	38.51	723.18	551.63	3.85
Topic Discovery					
Discovering 1 Topic	236.86	32.01	348.88	178.95	3.20
Discovering 10 Topics	378.33	33.42	548.11	32..68	3.34
Discovering 100 Topics	829.69	73.61	1057.50	712.84	7.36

Overview of NaradaBrokering

Multiple Transport Support: Transport protocols supported include TCP, Parallel TCP streams, UDP, Multicast, SSL, HTTP and HTTPS

Subscription Formats: Subscription constraints can be expressed as Strings, Integers, **XPath** queries, **Regular Expressions**, **SQL** and tag=value pairs

Messaging Related Compliance: Java Message Service **(JMS**) 1.02b compliant, **WS-Eventing** support.

Reliable Delivery: **Robust** and **exactly-once delivery** of messages in the presence of failures

Ordered Delivery: **Producer Order** and **Total Order** over a message type. **Time Ordered** delivery using Grid-Wide **NTP-based absolute time**

Recovery and Replay: **Recovery from failures** and disconnects. **Replay** of messages while preserving time-spacing between successive messages. **Buffering** services to reduce **Jitter**.

Security: **Secure end-to-end delivery of messages**

Message Payload Options: **Compression** and **Decompression** of payloads. **Fragmentation** and **Coalescing** of payloads.

Web Services: **WS-Eventing**, **WS-Reliable Messaging** and **WS-Reliablility**

Conclusions

- Provenance: We used this in our security framework to enforce dissemination authorizations and the corresponding durations for these rights.
  - Was used to cope with denial of service attacks.
- Life cycle management: Topics can be garbage collected.
- Discovery: May be restricted to the possession of valid credentials.