



IBM Software Group

WS-Notification overview

Peter Niblett

WebSphere. software



 e-business software

The Notification Pattern

- Consumer Registration
 - ▶ Entities interested in receiving information register with entities capable of producing information
- Multiple consumers
 - ▶ There may be many entities registered to receive the same information
- Multiple pieces of information
 - ▶ Once registered, a consumer may receive multiple messages
- Information is delivered asynchronously to the consumer



Goals

1. Provide WS-* standardisation of the asynchronous notification pattern
 - ▶ Concepts and Terminology
 - ▶ Standard Message Exchanges(PortTypes)
 - ▶ XML and WSDL rendering
2. Provide WS-* standardisation of a Notification Broker
3. Provide standard language to name and describe Topics

... in order to achieve

- ▶ Interoperation between independent publishers and consumers
- ▶ Interoperation between middleware providers
- ▶ Standardized taxonomy for Topics
- ▶ Standardised concepts and terminology



Terminology

- Situation
 - ▶ Some occurrence within a Web service or its environment of interest to third parties
- NotificationMessage
 - ▶ An artefact of a Situation containing information about that event that some entity wishes to communicate to other entities
 - ▶ Represented as an XML element with a QName with an XML-Schema defined type.
- NotificationProducer
 - ▶ A Web service capable of distributing messages. It implements the NotificationProducer message exchanges and supports one or more Topics
- NotificationConsumer
 - ▶ A Web service that receives notification messages from a NotificationProducer
- Publisher
 - ▶ An entity that creates Messages, it need not itself be a web service.
 - ▶ it may be a NotificationProducer, or it may use the services of a NotificationBroker

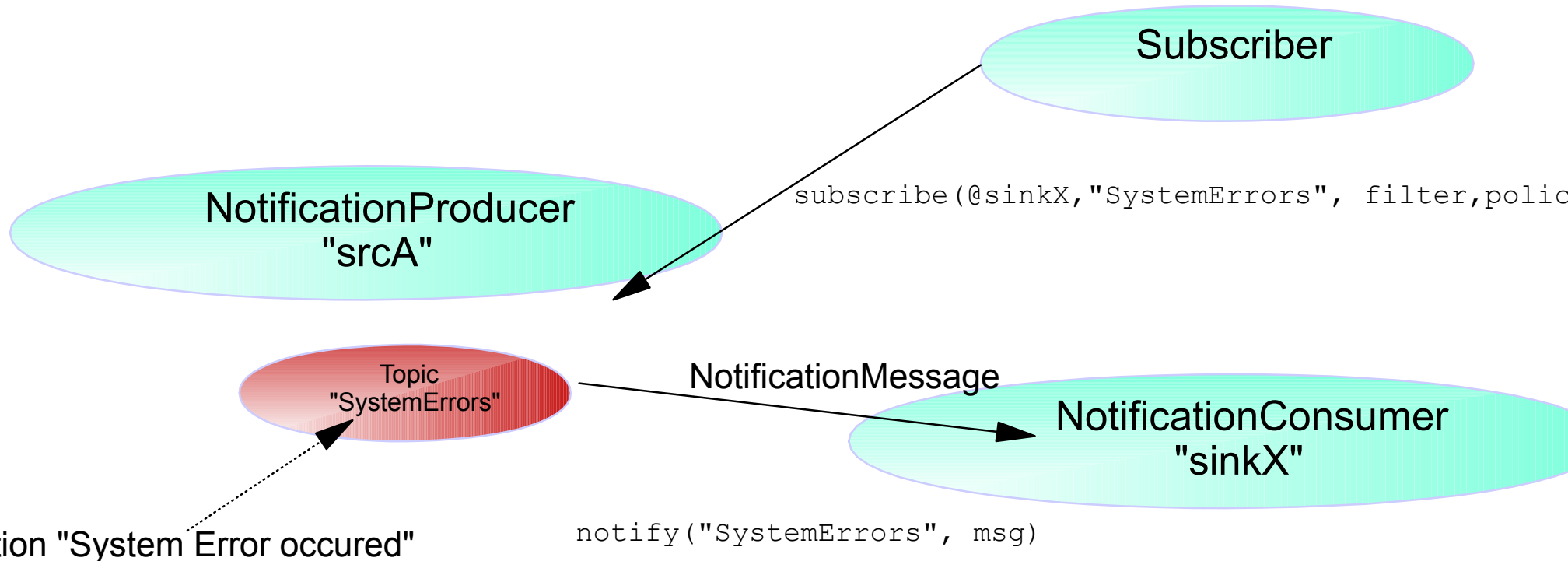


Non Goals

- Prescribing the format of messages used to pass event information
- Defining any particular "standard Events" or "standard Messages"
- Specifying the mapping between Situations and Messages
- Defining the means by which NotificationProducers and NotificationBrokers are discovered by Subscribers
- Defining a specific Policy language for notification



Basic Pattern



Situation "System Error occured"

`notify("SystemErrors", msg)`

Subscriptions (managed by SubscriptionManager)

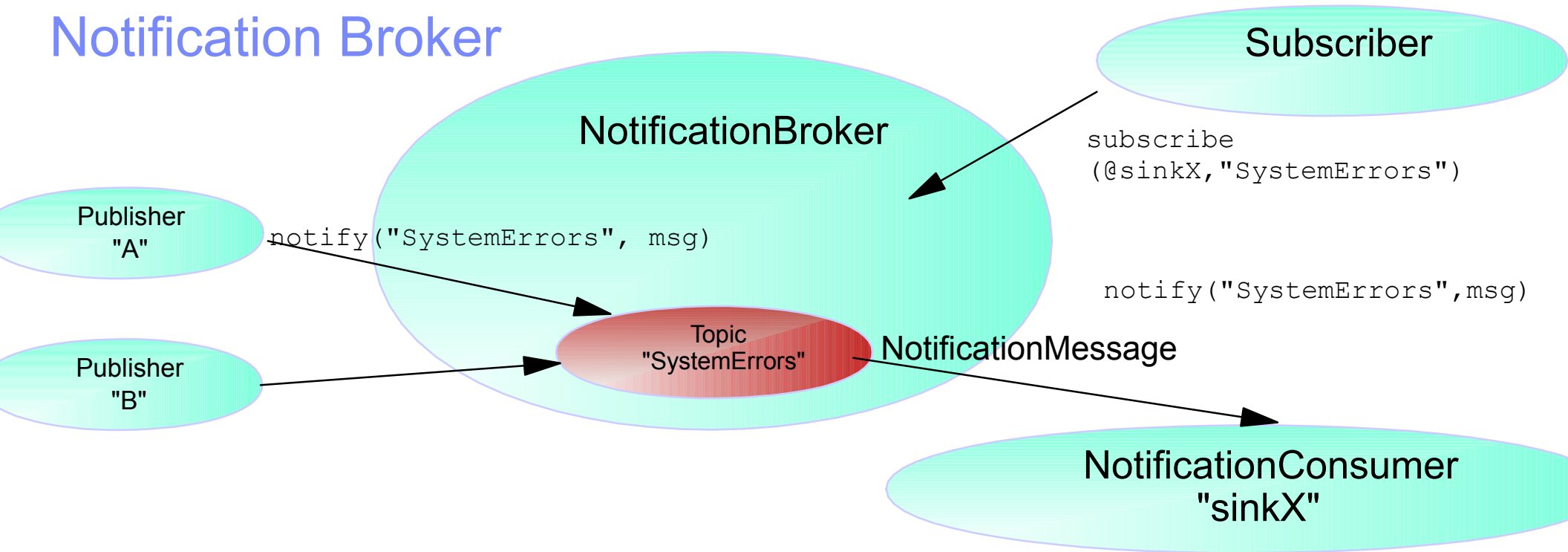
{ @sinkX, "SystemErrors", filter, ... }

Explicit or time-based deletion of subscriptions

@xxx means "Endpoint reference to xxx"



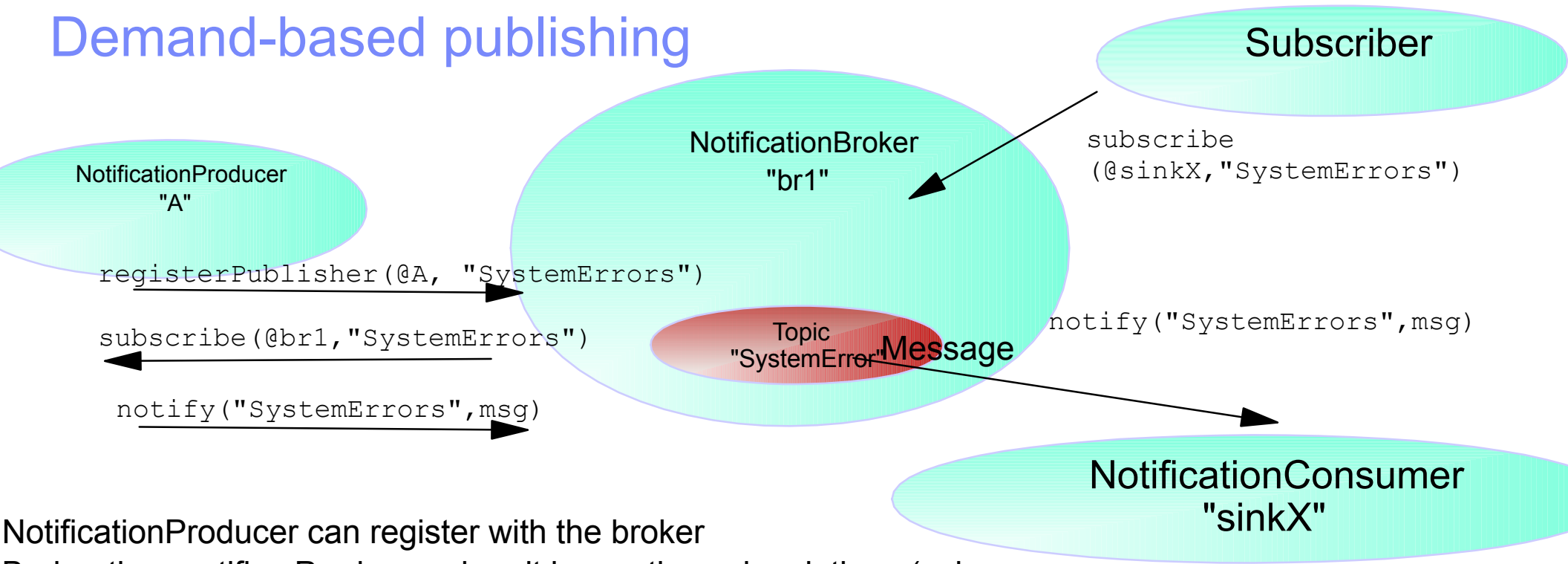
Notification Broker



- NotificationBroker implements NotificationProducer and NotificationConsumer operations
- It manages subscriptions on behalf of publishers (scalability and separation of concerns)
- Brokered configurations reduce the number of interconnects (scalability)
- Point to administer and control subscriptions
- Subscribers do not need to discover individual publishers
- Can hide the identity of the publisher



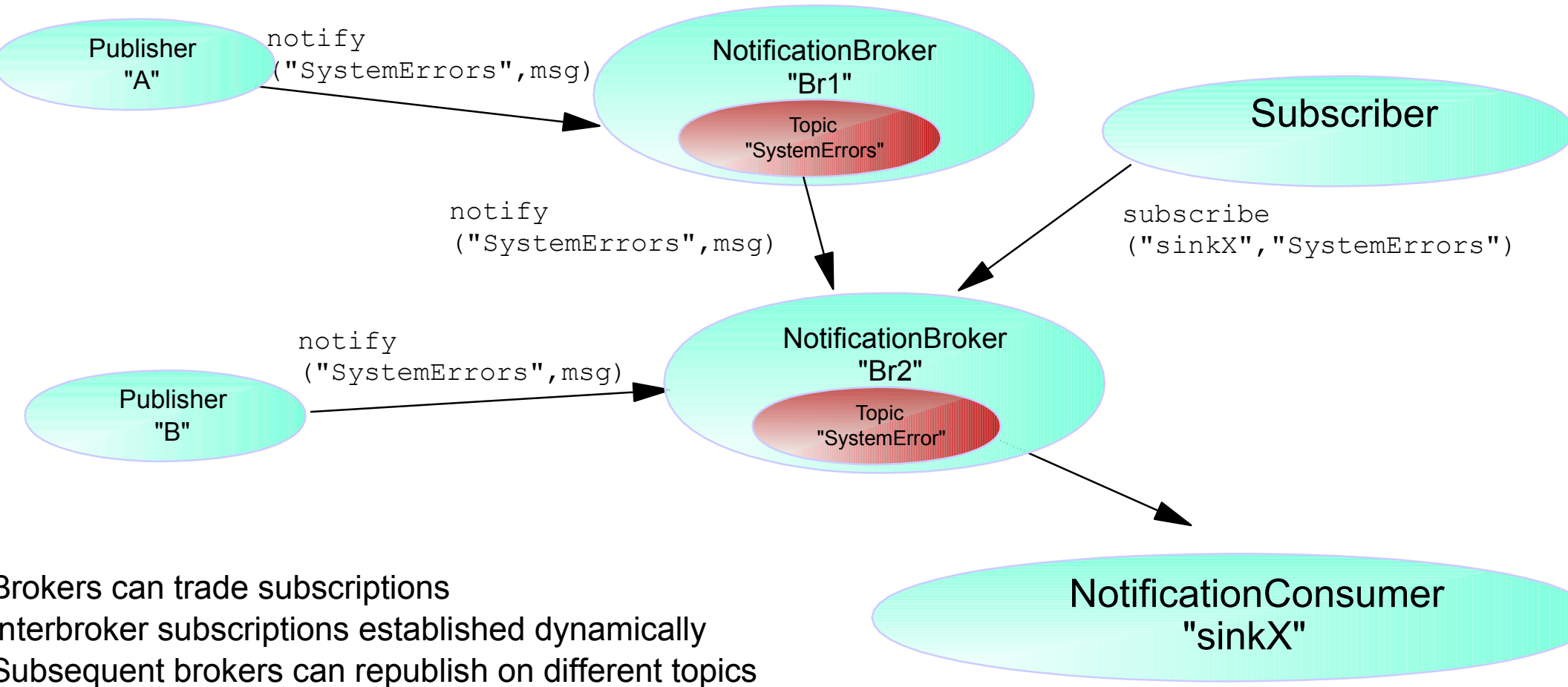
Demand-based publishing



NotificationProducer can register with the broker
 Broker then notifies Producer when it has active subscriptions (using an aggregated subscription)
 Avoids potentially costly production of messages if there are no subscribers



Federated Brokers



Brokers can trade subscriptions
 Interbroker subscriptions established dynamically
 Subsequent brokers can republish on different topics



Topics

- Method of categorising Notifications
 - ▶ Each Topic is associated with one or more MessageSchemas
 - ▶ Grouped into namespaces which have URIs
 - ▶ Topics may be defined by an individual publisher, or by a 3rd party (e.g. a standards organisation)
 - ▶ Allow a potential subscriber to understand the notification capabilities of a Publisher/Broker that has just discovered.
 - ▶ Facilitate interoperation with Topic-based IT pub/sub systems (e.g. JMS)
 - ▶ Convenient attachment point for access control

- Subscriber provides a TopicPathExpression to indicate Topic(s) of interest
 - ▶ Matching on a Topic name is usually more efficient than matching against general message content.
 - ▶ Simple Topic Path Expression is just a single Topic Name
 - ▶ A simple NotificationSource may support only one Topic
 - ▶ More complex NotificationSources may permit wildcard expressions

