Chapter 4
Design Patterns

Publisher-Subscriber

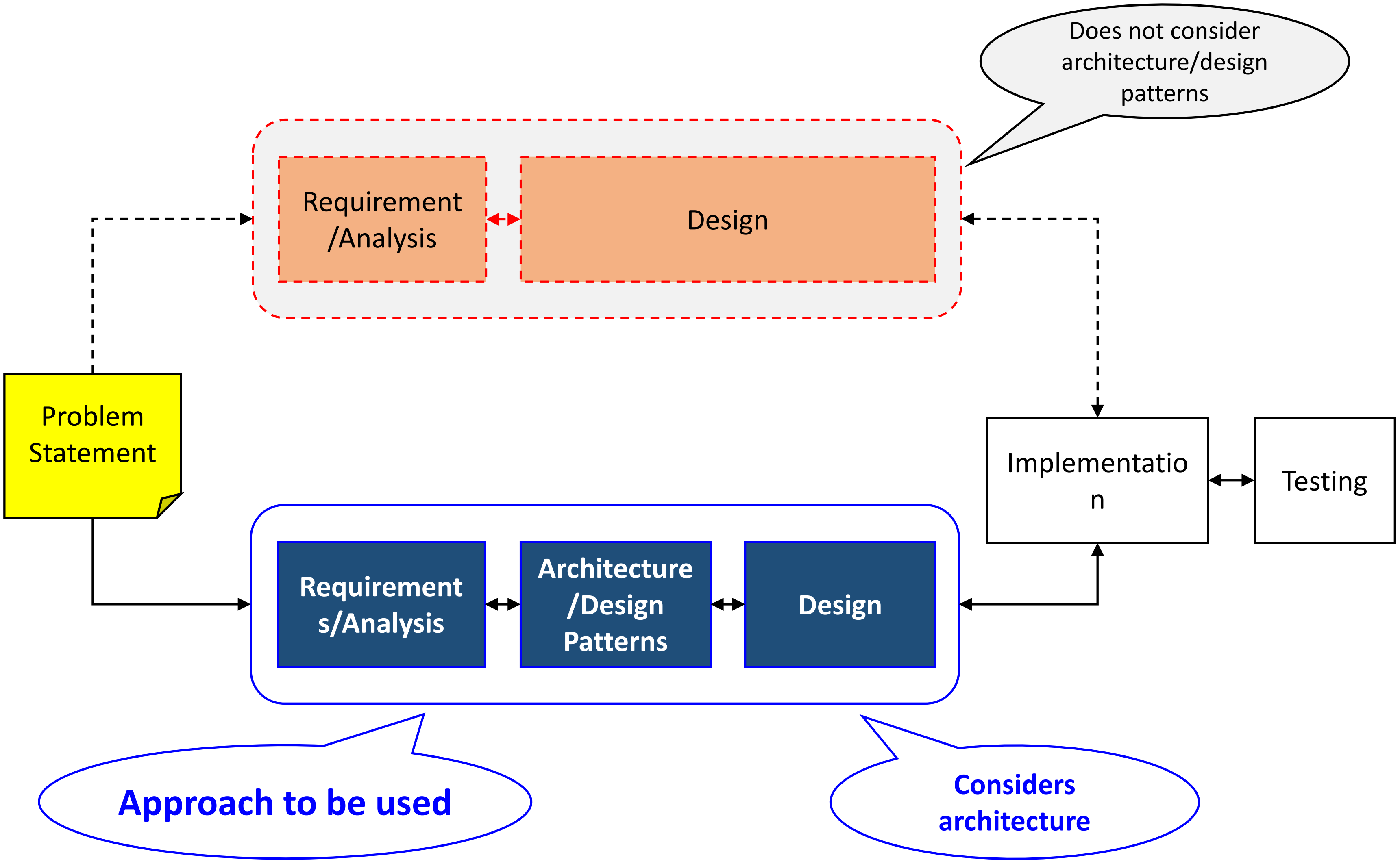
H.S. Sarjoughian

CSE 460: Software Analysis and Design

School of Computing, Informatics and Decision Systems Engineering
Fulton Schools of Engineering

Arizona State University, Tempe, AZ, USA

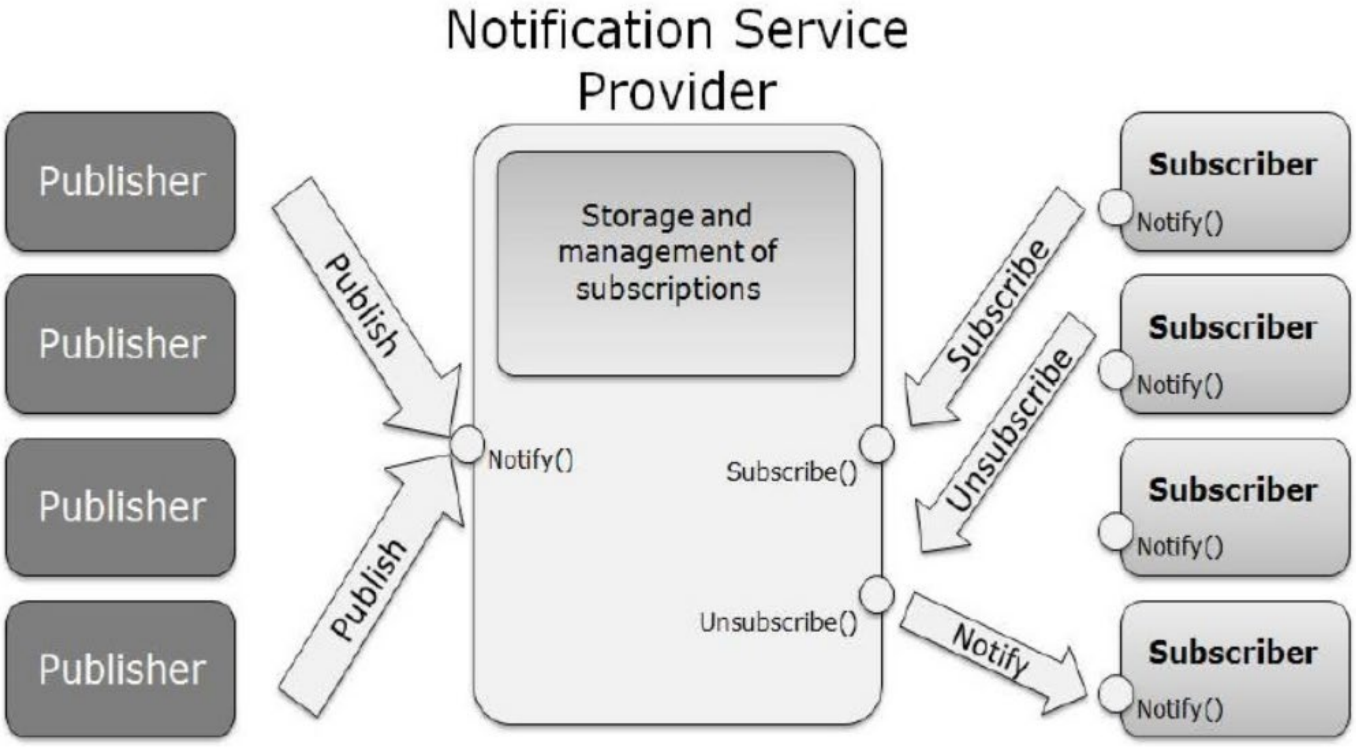
Software Engineering Process



Applicability

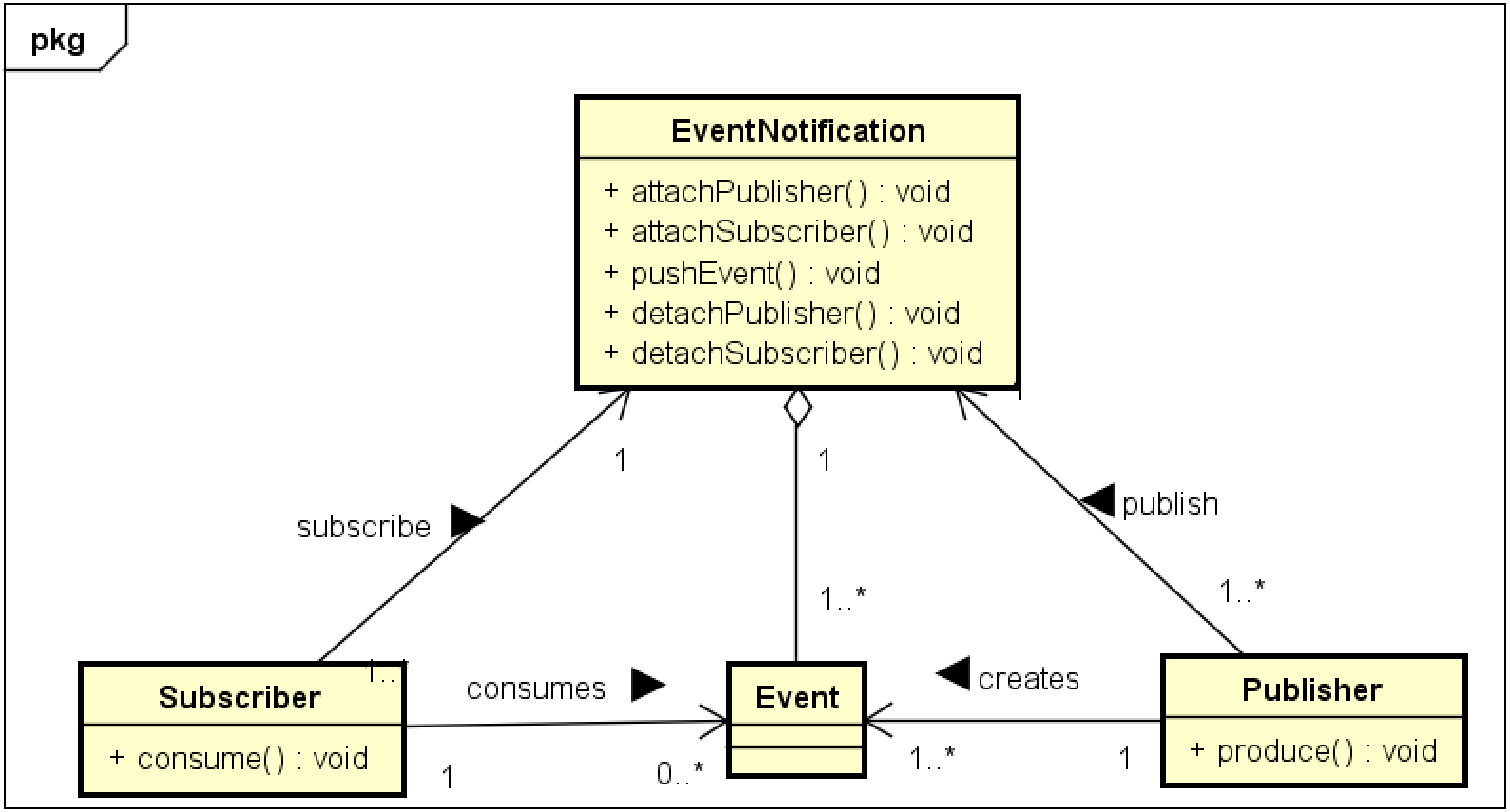
- The software system has two aspects where one is dependent on another
- Change in one object can result in any number of other objects to change
- An object is responsible for notifying any number of other objects
- Some objects are interested in receiving notifications

Structure and Participants



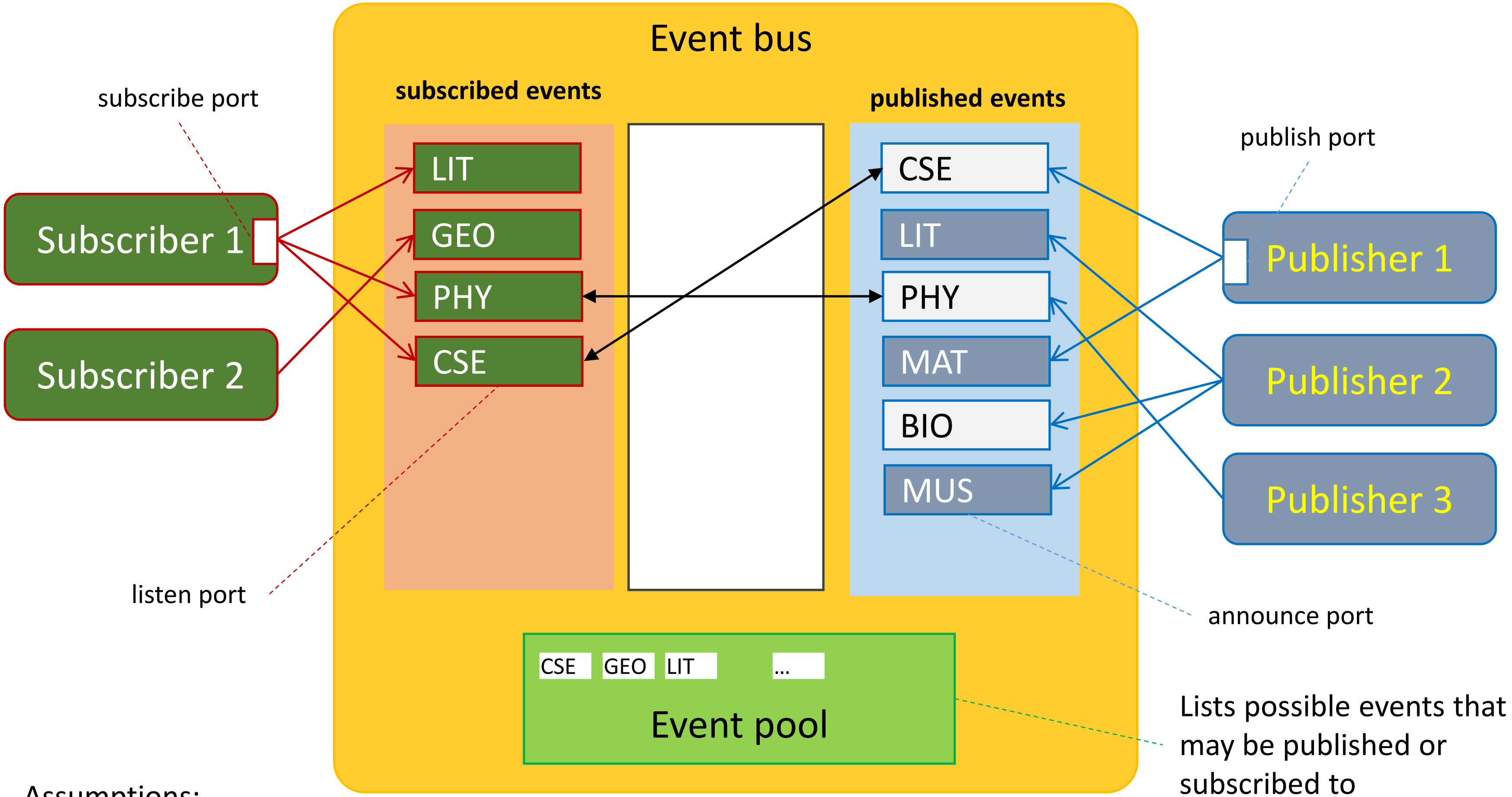
Credit: Reference [3]

Structure and Participants



Details of the methods (e.g., visibility, return type, and arguments) need to be specified for classes

Publisher-Subscriber Design Pattern Example

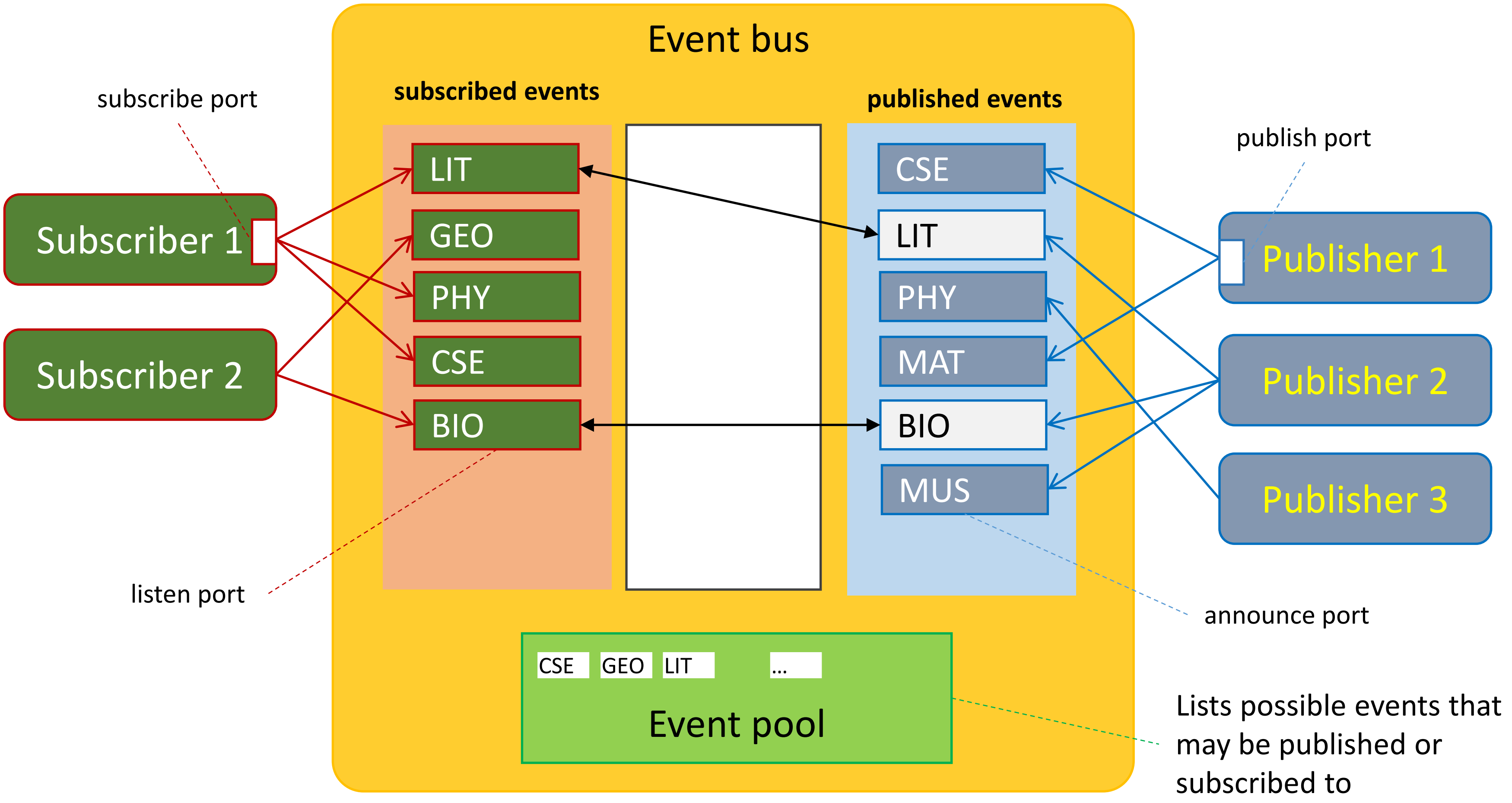


Lists possible events that may be published or subscribed to

Assumptions:

- Each component can be either publisher or subscriber
- Events (from both publisher and subscriber) are independent and totally-ordered
- Number of publishers, subscribers, and events are arbitrary

Publish-Subscribe Design Pattern Example (cont.)



Lists possible events that may be published or subscribed to

Consequences and Known Uses

- Advantages
  - Modularity: publishers and subscribers are unaware of each other (publishers and subscribers can vary independently)
  - Extensibility: scalable (publishers and subscribers are simpler to add and remove)
  - Customizability: subscribers can do different things
- Disadvantages
  - Subscribers and publishers may affect the overall system in undesirable ways since subscribers (and publishers) don't know anything about their siblings.
- Some uses
  - Publishers/subscriber middleware
  - Cloud software systems (e.g., Dropbox)

References

[1] Pattern-Oriented Software Architecture, Volume 4, A Pattern Language for Distributed Computing, F. Buschmann, K. Henney, D. C. Schmidt, Wiely, 2007

[2] Ishraq Fataftah, 2012. “Publish Subscribe model overview” Available at: https://www.slideshare.net/RobertoBaldoni/distributed-event-routing-in-publishsubscribe-systems?next_slideshow=1

[3] Software Architecture in Practice, 3rd Edition, (SAP3), Bass, L. Clements, P., and Kazman, R., Addison Wesley, SEI Series in Software Engineering, 2012