

Quiz

Now videos...

CS4501

Robotics for Soft Eng

...

Features Of Software Development for Robots

SW Specification Differences - Trickleing of Physical World

- State Properties
 - Rate of descent < 3m/s
 - Angle < 17 degrees
 - Calibrated = True

Properties may include
physical terms

SW Specification Differences - Trickle of Physical World

- State Properties
 - Rate of descent $< 3\text{m/s}$
 - Angle < 17 degrees
 - Calibrated = True
- Conditional State Properties
 - If approaching, then speed $< \text{delta}$
 - If taking off, proximity sensor should be false

Properties may include physical terms

Properties are state-dependent

SW Specification Differences - Trickle of Physical World

- State Properties
 - Rate of descent < 3m/s
 - Angle < 17 degrees
 - Calibrated = True
- Conditional State Properties
 - If approaching, then speed < delta
 - If taking off, proximity sensor should be false
- Timeliness properties
 - Frequency Heartbeat = 20hz
 - Abort sequence takes less than 2s
- Temporal properties
 - Battery > 30% before Takeoff
 - Translation can only occur after takeoff

Properties may include physical terms

Properties are state-dependent

Timeliness matters

SW Architectural and Design Differences

- Asynchronous
- Loosely coupled
- Abstracted
- Close-loop

SW Architectural and Design Differences

- Asynchronous, event-driven -- world operates that way
- Loosely coupled -- parallelization, reuse
- Abstraction -- manage complexity
- Close loop -- need to assess/respond to changes

SW Differences: Publish/Subscribe

Publisher

Sends messages

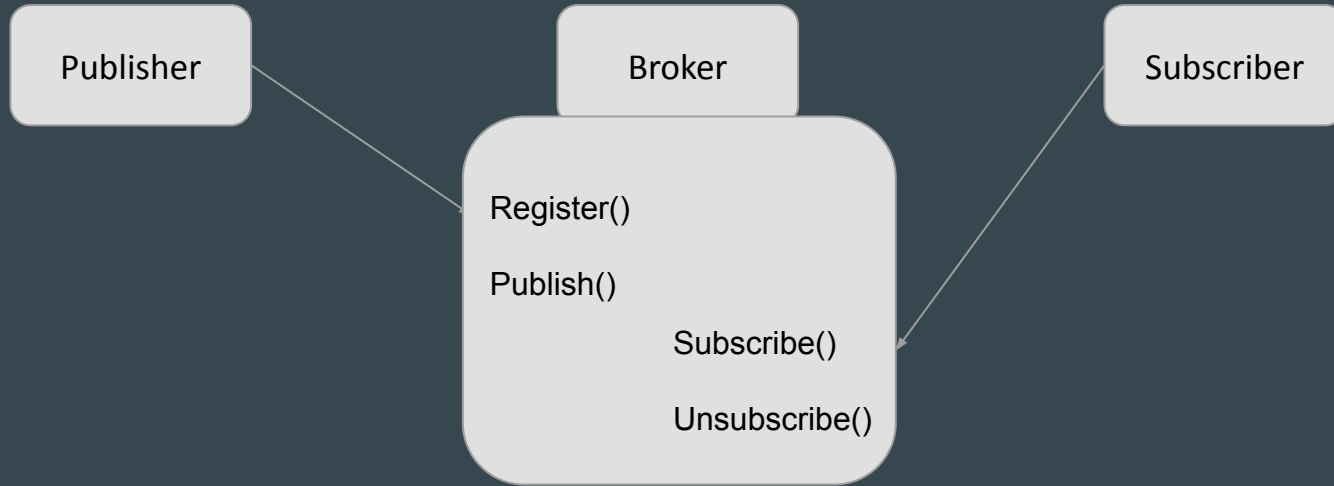
Broker

Manages
subscriptions,
publishers, filtering,
and sometimes
delivery

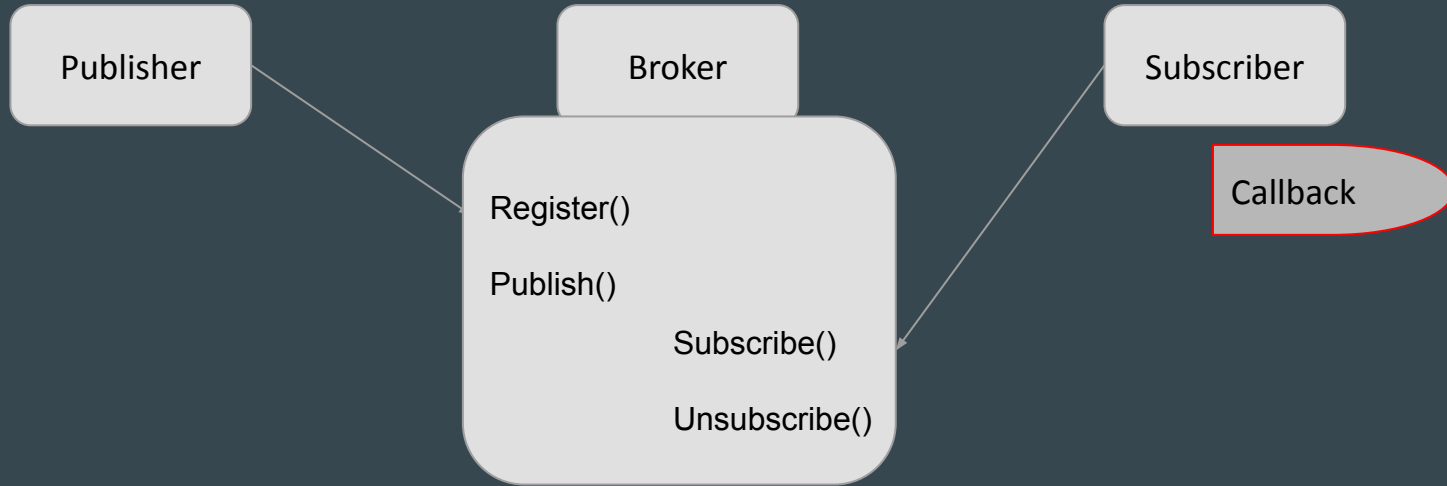
Subscriber

Receives messages

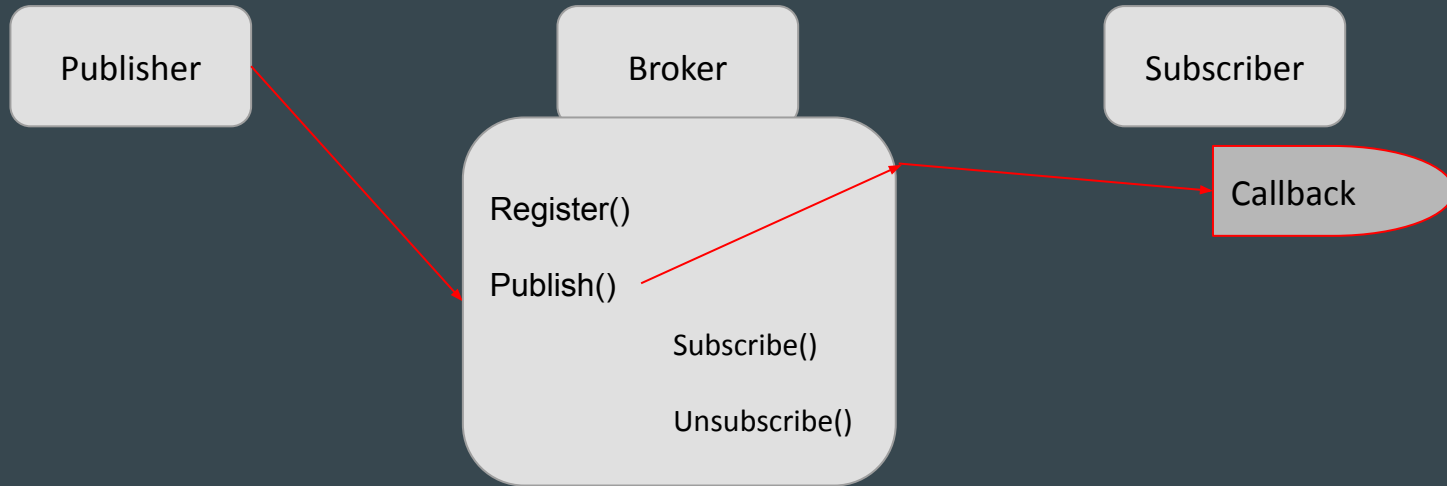
SW Differences: Publish/Subscribe



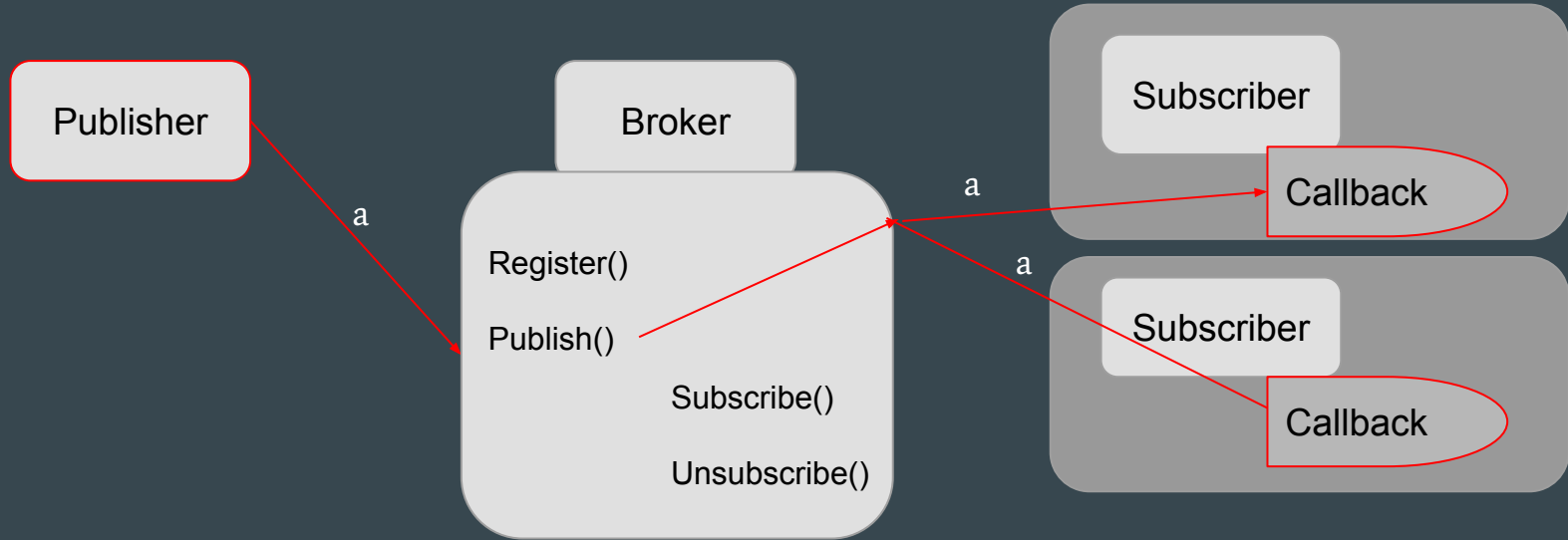
SW Differences: Publish/Subscribe



SW Differences: Publish/Subscribe

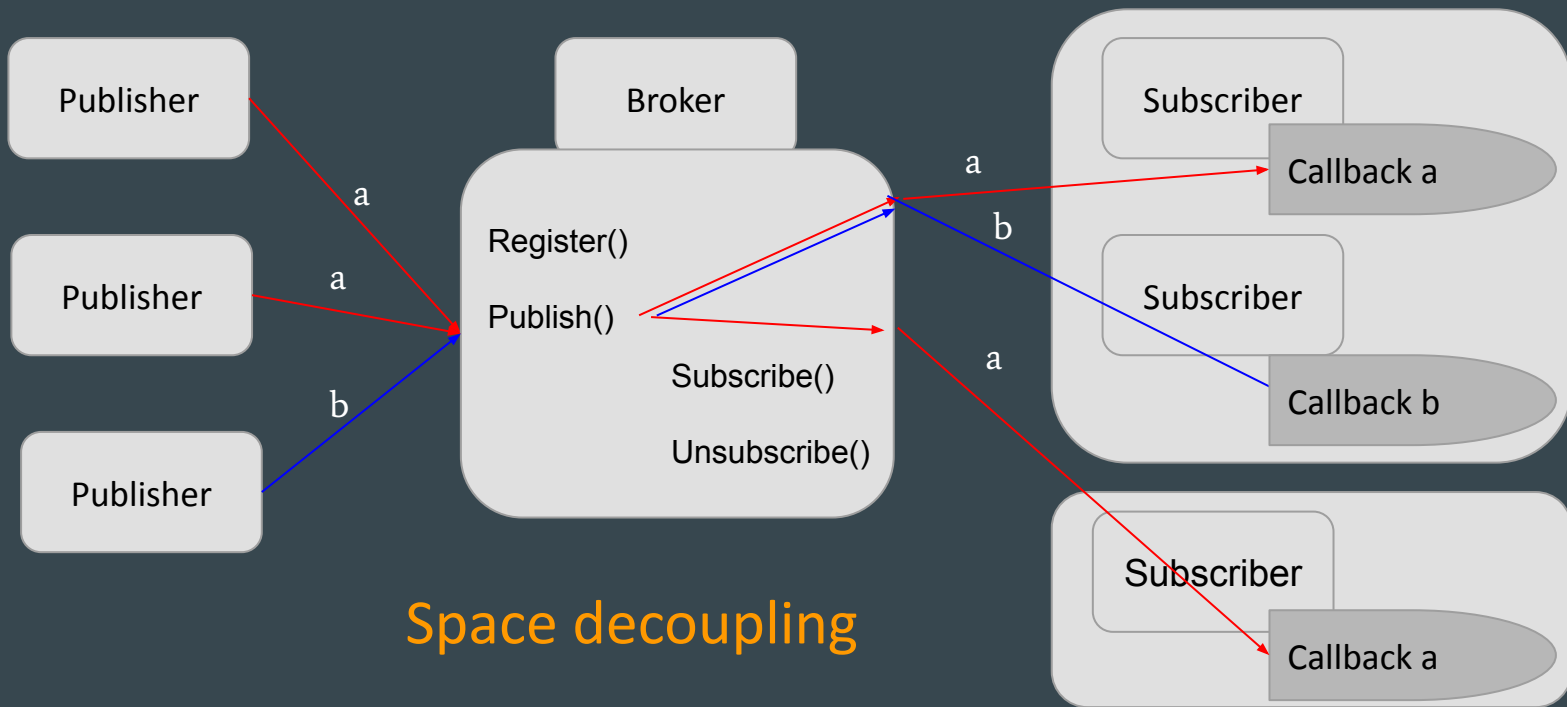


SW Differences: Publish/Subscribe

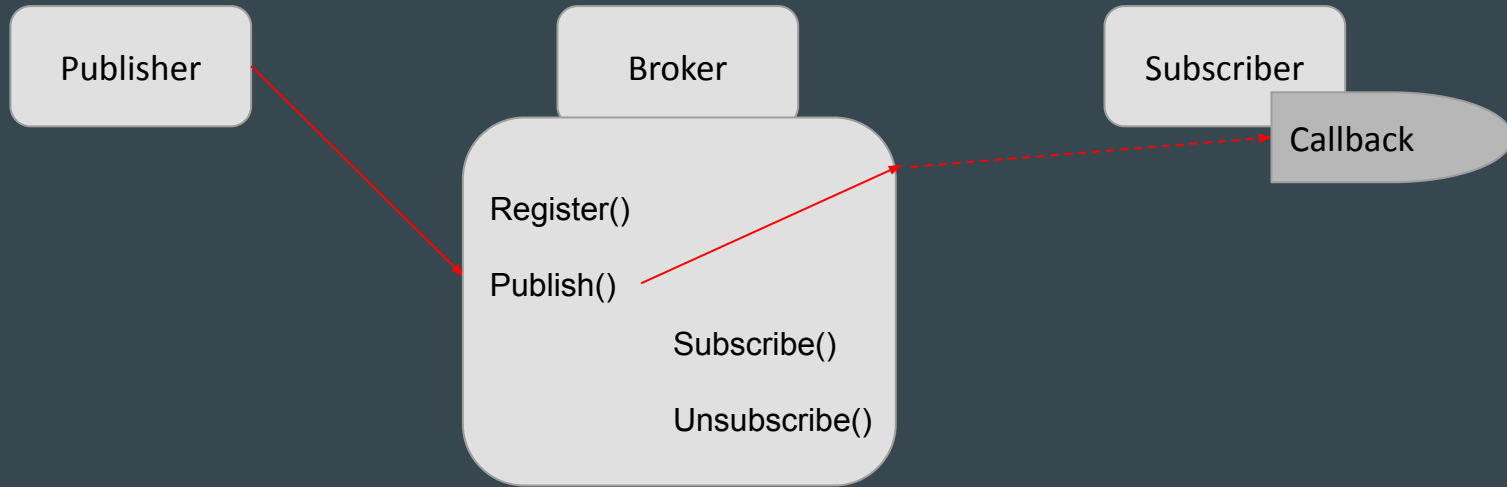


Space decoupling

SW Differences: Publish/Subscribe

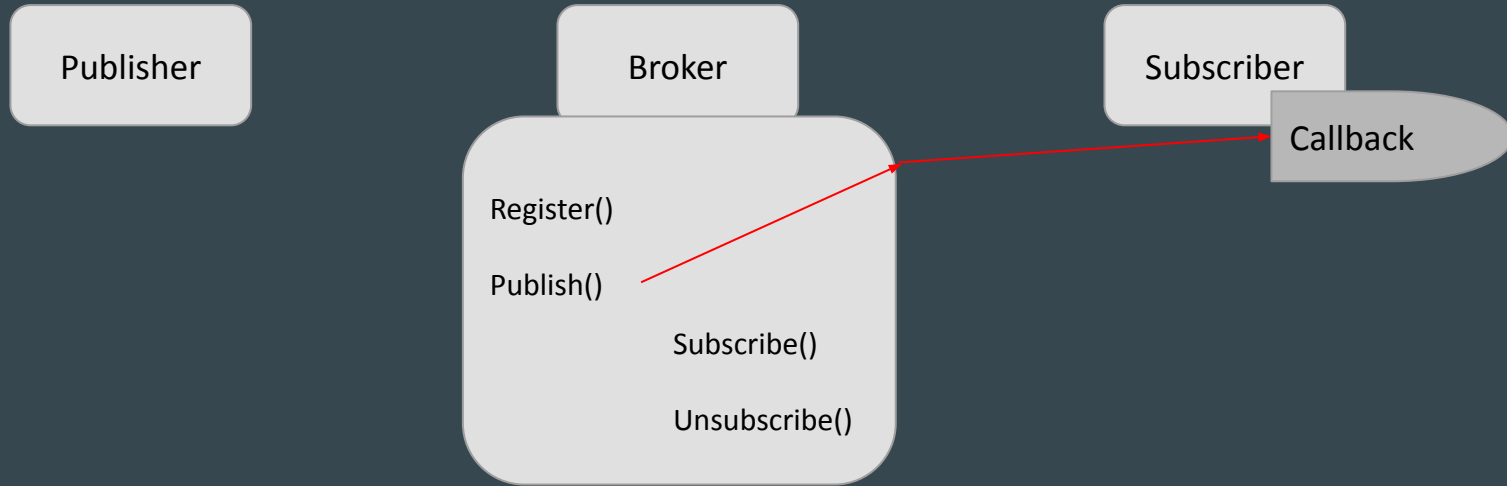


SW Differences: Publish/Subscribe



Time decoupling

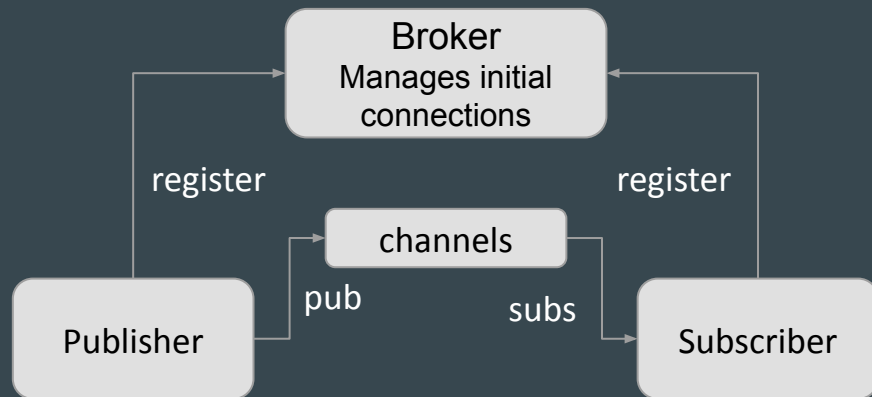
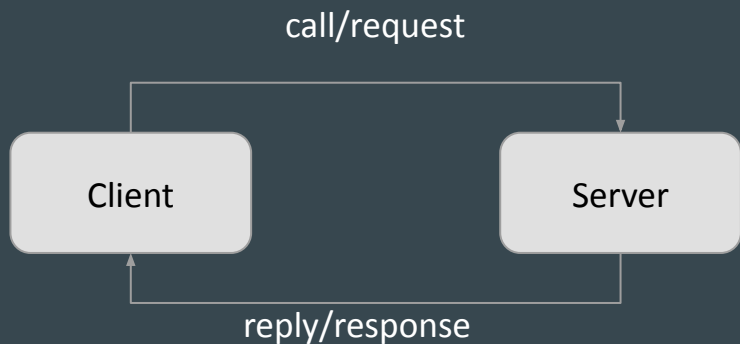
SW Differences: Publish/Subscribe



Time decoupling

Publish/Subscribe vs. Client-Server

Synchronous

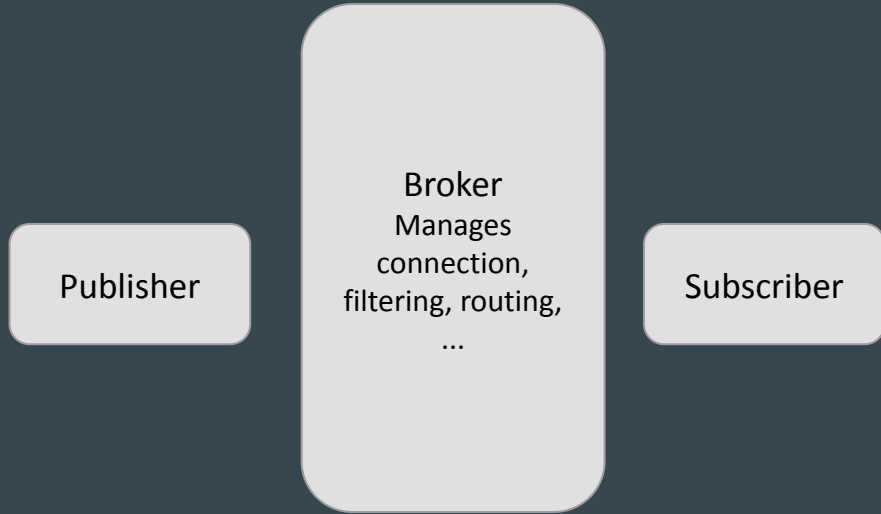


Asynchronous

SW Differences: Publish/Subscribe Functionality

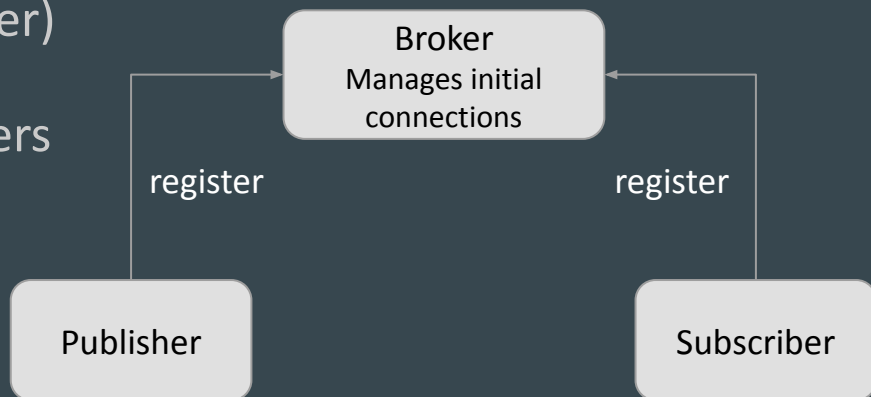
- Filtering
 - Which subscribers get what messages
 - Topic-based
 - Content-based
- Routing
 - Getting those messages to subscribers
 - Alternatives: Unicast / Multicast / Push-pull

SW Differences: Publish/Subscribe ROS



SW Differences: Publish/Subscribe ROS

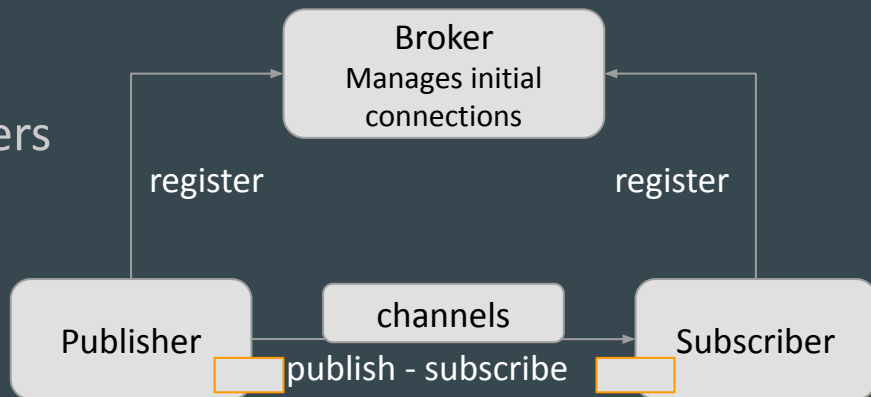
- Broker is a Manager (core+param server)
- Nodes can be Publishers and Subscribers



ROS also offers Client-Server, and Actions

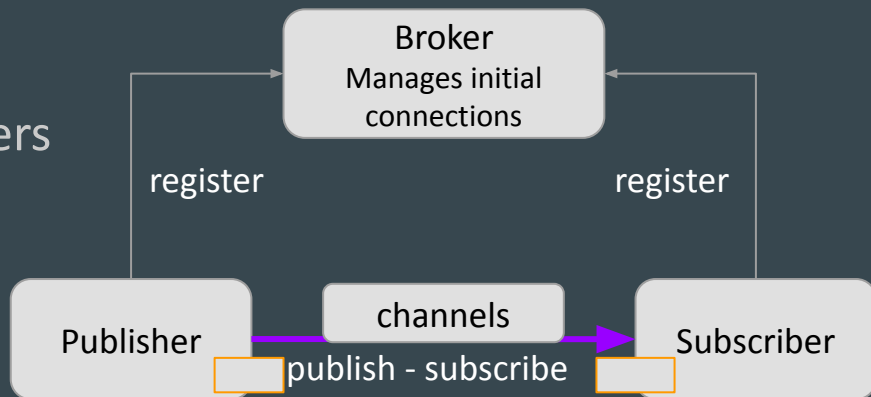
SW Differences: Publish/Subscribe ROS

- Broker is a Manager
- Nodes can be Publishers and Subscribers
- **Topic-based** filtering with **buffering**



SW Differences: Publish/Subscribe ROS

- Broker is a Manager
- Nodes can be Publishers and Subscribers
- Topic-based filtering with buffering
- Peer-to-peer routing



SW Differences: Publish/Subscribe ROS

- Broker is a Manager
- Nodes can be Publishers and Subscribers
- Topic-based filtering with buffering
- Peer-to-peer routing
- Standardized common msg formats



SW Design Differences: Publish/Subscribe

- Broker is a Manager
- Nodes can be Publishers and Subscribers
- Topic-based filtering with buffering
- Peer-to-peer routing
- Standardized common msg formats

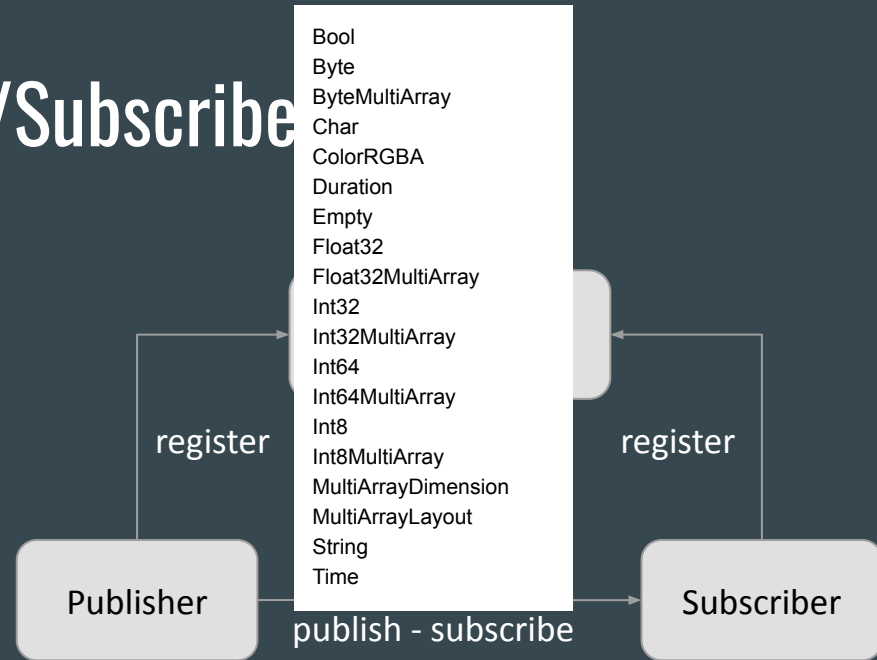


* Accel
* AccelStamped
* AccelWithCovariance
* AccelWithCovarianceStamped
* Inertia
* InertiaStamped
* Point
* Point32
* PointStamped
* Polygon
* PolygonStamped
* Pose
* Pose2D

SW Design Differences: Publish/Subscribe

- Broker is a Manager
- Nodes can be Publishers and Subscribers
- Topic-based filtering with buffering
- Peer-to-peer routing
- Standardized common msg formats

- Standard types
- State
- Sensors
- Actuators
- Navigation

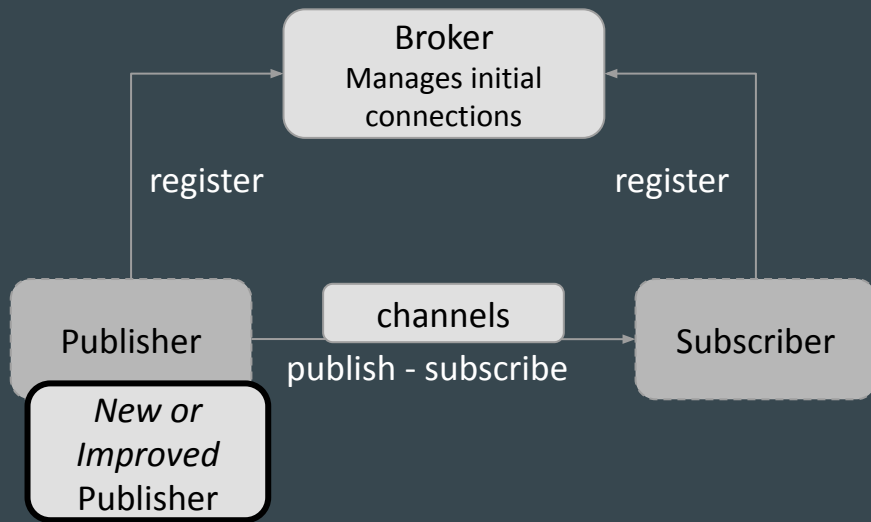


- BatteryState
- CameraInfo
- ChannelFloat32
- CompressedImage
- FluidPressure
- Illuminance
- Image
- Imu
- JointState
- Joy
- JoyFeedback
- * Accel
- * AccelStamped
- * AccelWithCovariance
- * AccelWithCovarianceStamped
- * Inertia
- * InertiaStamped
- * Point
- * Point32
- * PointStamped
- * Polygon
- * PolygonStamped
- * Pose
- * Pose2D
- GridCells
- MapMetaData
- OccupancyGrid
- Odometry
- Path

SW Design Differences: Publish/Subscribe ROS

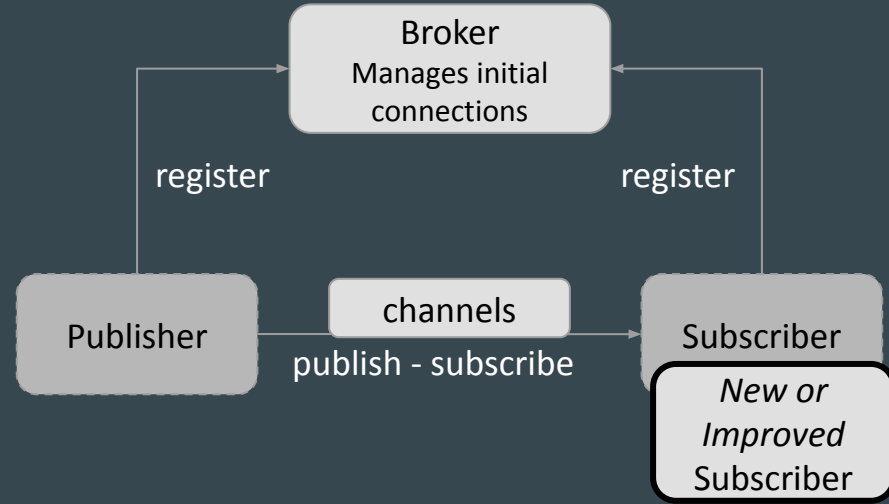
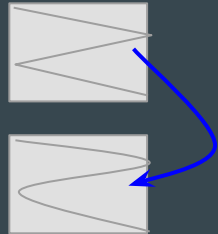


| Supported GPS and/or Compass | | |
|---|-----|------------------|
| Device | GPS | Compass |
| Hobbyking u-blox Neo-M8N GPS with Compass | MBN | ✓ |
| mRo GPS u-blox Neo-M8N Dual Compass | MBN | LIS3MDL, IST8310 |
| Drotek DP0804 (and other Drotek u-blox GPS/Compasses) | MBN | LIS3MDL |
| Emild Reach Mv12 - PX4 only supports "ordinary" GPS with this module. RTK support is expected in the near future. | ✓ | ✗ |
| Espressu M300 MBN GPS | MBN | IST8310 |



Plug-Play REUSE

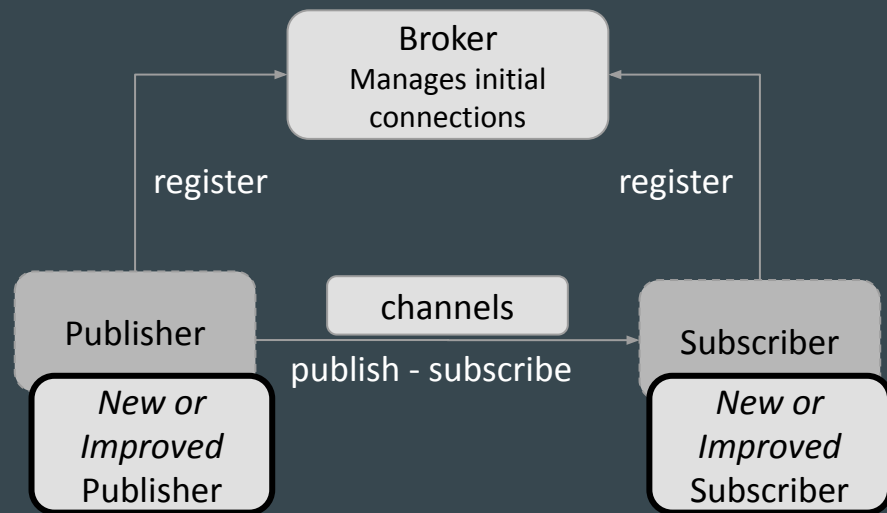
SW Design Differences: Publish/Subscribe ROS



Plug-Play REUSE

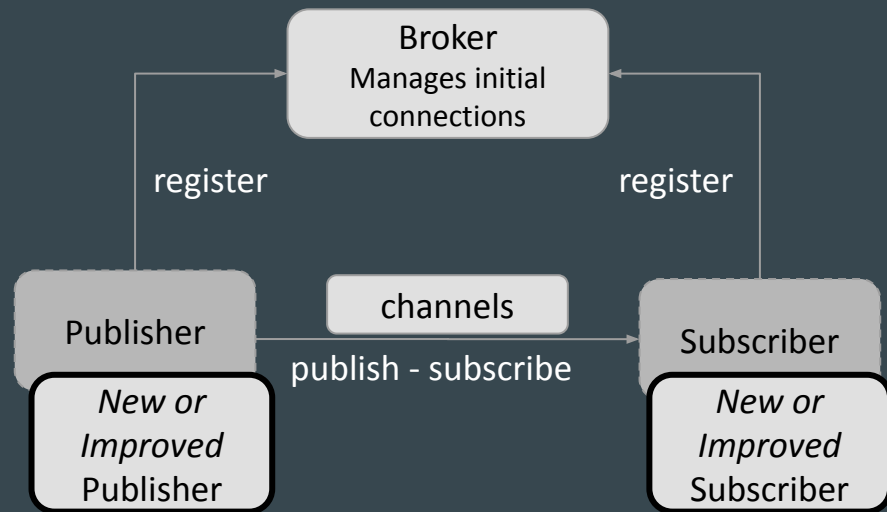
Law of Leaky Abstractions - ROS Pub/Sub

How can it leak?



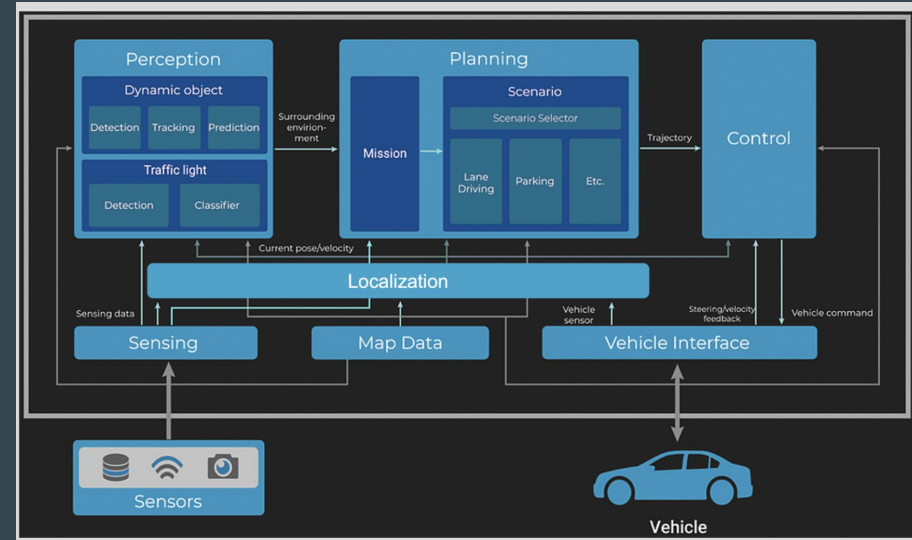
Law of Leaky Abstractions - ROS Pub/Sub

- Not everyone is bound to use standard messages or use them in the same way
- Channels do not offer real-time guarantees
- When underlying core hangs the results are unknown
- Size of the buffer matters
 - Implication of being too short?
 - Implication of being too long?



ROS

“**Autoware** is the world’s leading open-source software project for autonomous driving. Autoware is built on **Robot Operating System (ROS)** and enables commercial deployment of autonomous driving in a broad range of vehicles and applications”



Takeaways

- More complex development process, branch / sync / integrate
- Richer specifications that must include the physical world
- Many abstractions, many of them Leaky
- Simulation is a big part of modeling and testing
- Programming the deployment
- Asynchronous, event-driven, loosely coupled architectures
- Publish/Subscribe architecture, P/S ROS