Features of Software Development for Robots

CSCI 420-04 Robotics

WILLIAM & MARY

CHARTERED 1693

Software Specification: Physical

- State Properties:
  - Acceleration < $4.9 \frac{m}{s^2}$
  - Altitude < 35,000 ft
  - Battery > 80% charge
  - Calibrated == True
  - Located within Washington, D.C.

Properties can include **physical terms** and/or relate to the **physical world**

Software Specification: Dependent

- Conditional State Properties
  - If descending, landing gear out
  - If turning, speed $< 3 \frac{m}{s}$
  - If raining, headlights must be on
  - If downhill, low gear must be engaged

Properties can depend on **world** and **system** state

Software Specification: Timeliness

- Timeliness properties
  - (not robot) The cabin crew must be capable of evacuating all passengers in 90 seconds.
  - Brake must engage within 20ms of signal
  - Watchdog timer must run at 50Hz

Ability to respond may be impacted by physical world

Software Specification: Temporal

- Temporal Properties
  - Engine must ignite **before** launch
  - Car must wait at stop sign **until** its turn
  - Drone must **eventually** return to base

Robot SW Architecture

- Asynchronous
- Loosely Coupled
- Abstracted
- Closed-loop

Robot SW Architecture

- Asynchronous – Event Driven
- Loosely Coupled – Parallelization, reuse
- Abstracted – Manage Complexity
- Closed-loop – Respond to Change

Event-Driven Comms: Pub/Sub

Publisher

Sends Messages

Broker

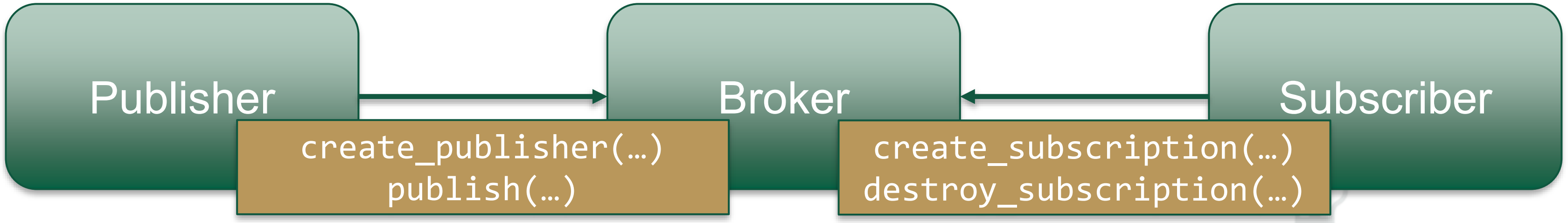
“Middleware”

Manages registry, data conversion, delivery.

Subscriber

Receives Messages

Event-Driven Comms: Pub/Sub



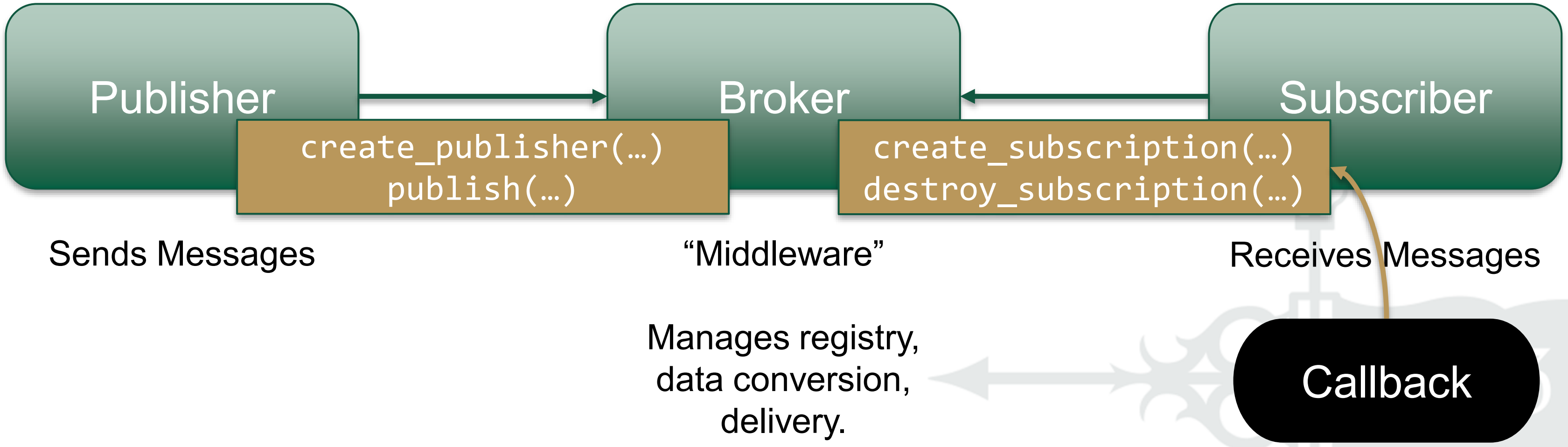
Sends Messages

“Middleware”

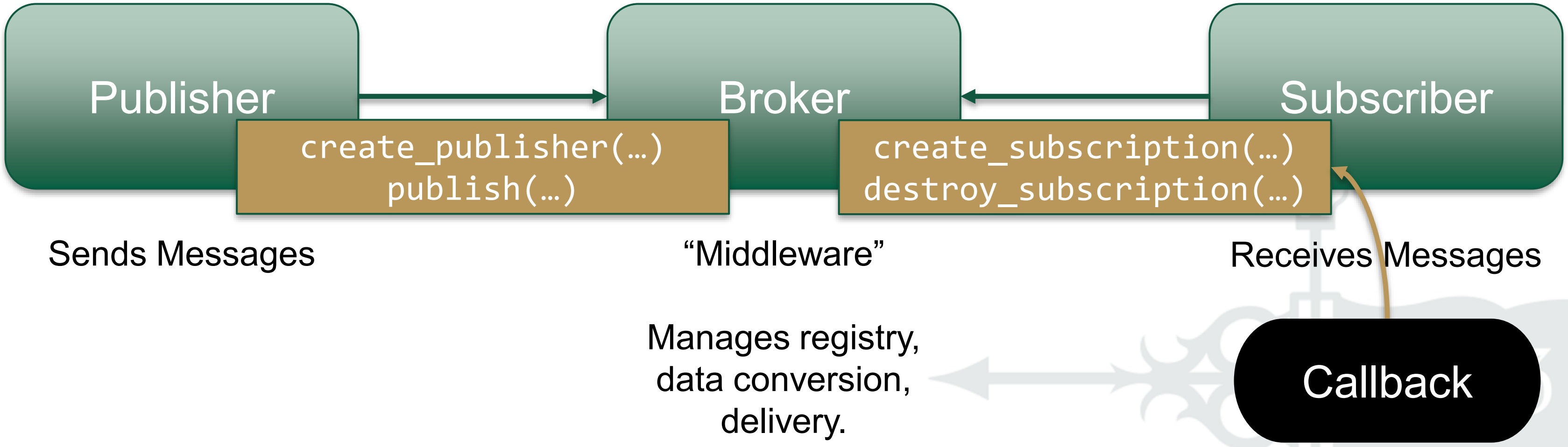
Manages registry, data conversion, delivery.

Receives Messages

Event-Driven Comms: Pub/Sub



Event-Driven Comms: Pub/Sub



```
create_publisher(…)
publish(…)
```

```
create_subscription(…)
destroy_subscription(…)
```

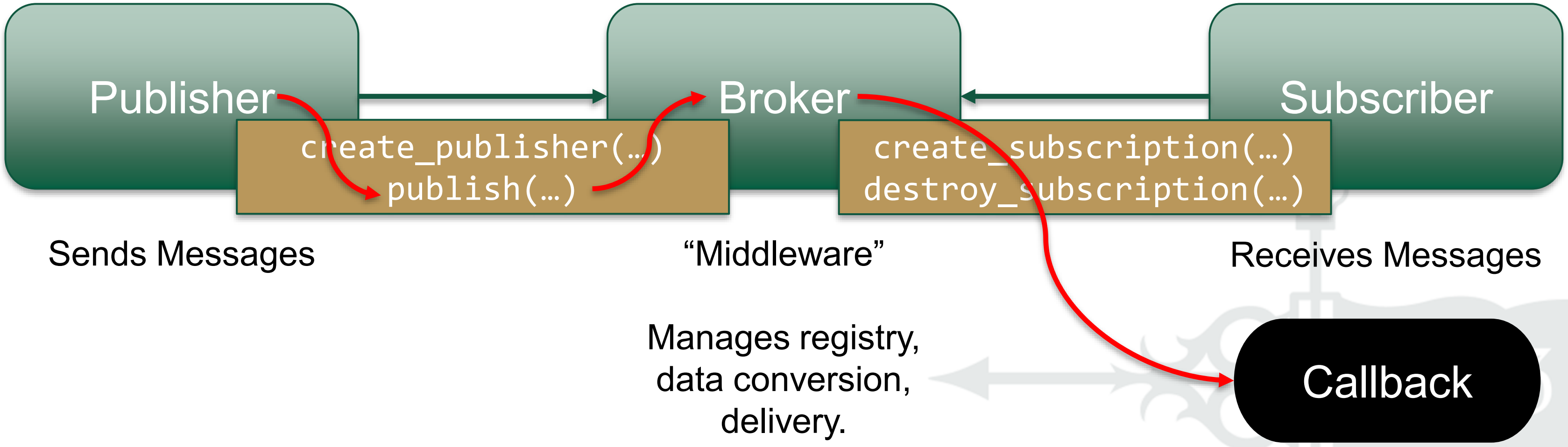
Sends Messages

“Middleware”

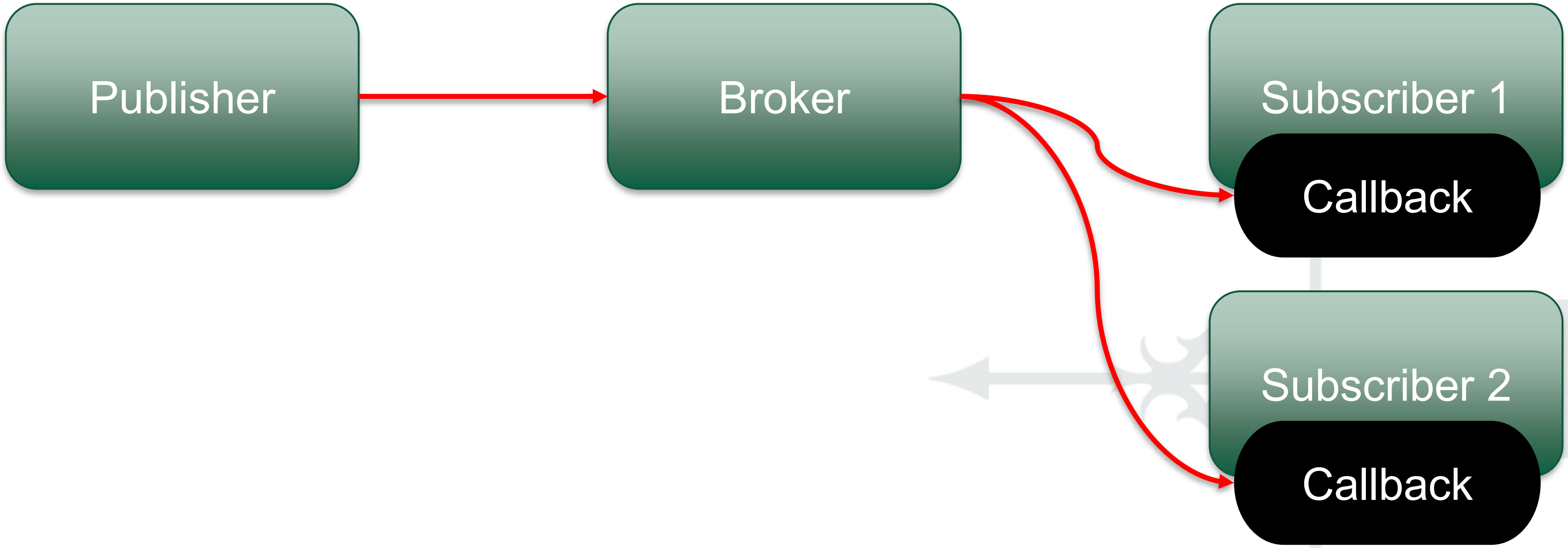
Manages registry, data conversion, delivery.

Receives Messages

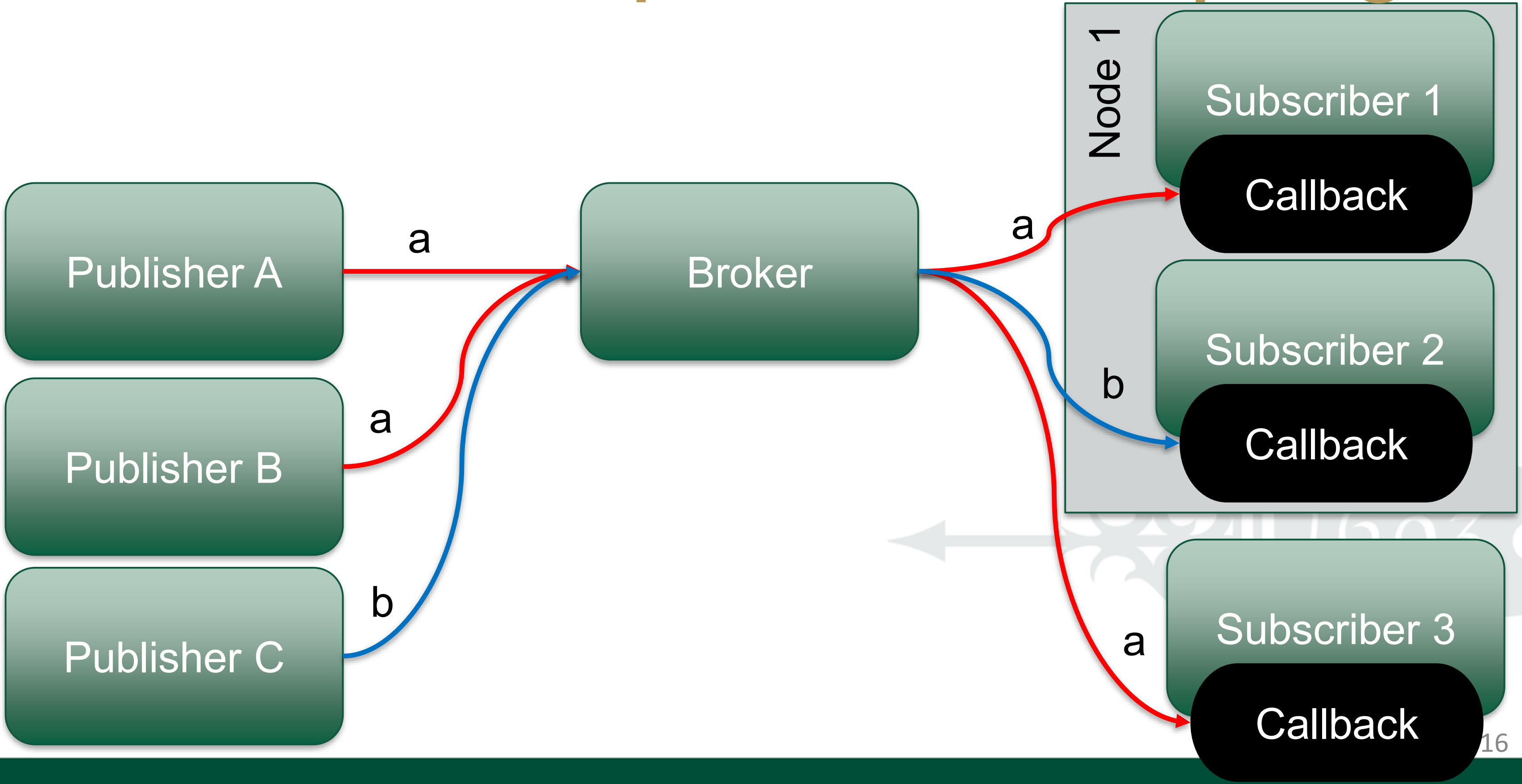
Event-Driven Comms: Pub/Sub



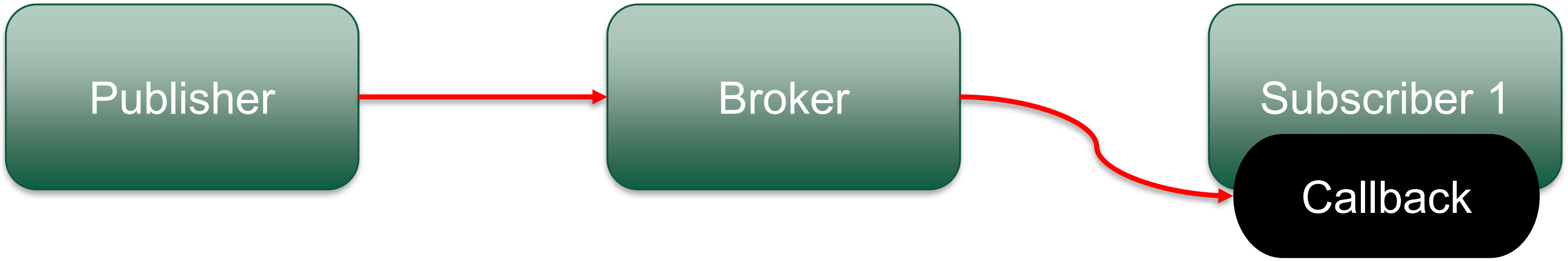
Pub/Sub: Space Decoupling



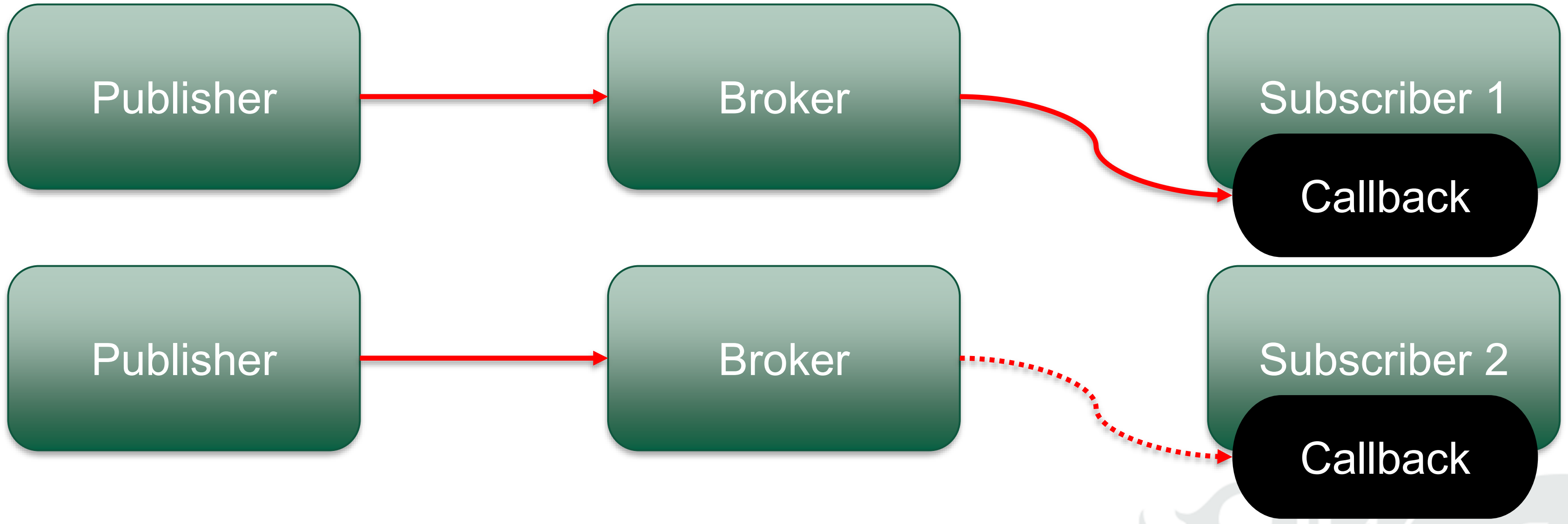
Pub/Sub: Space Decoupling



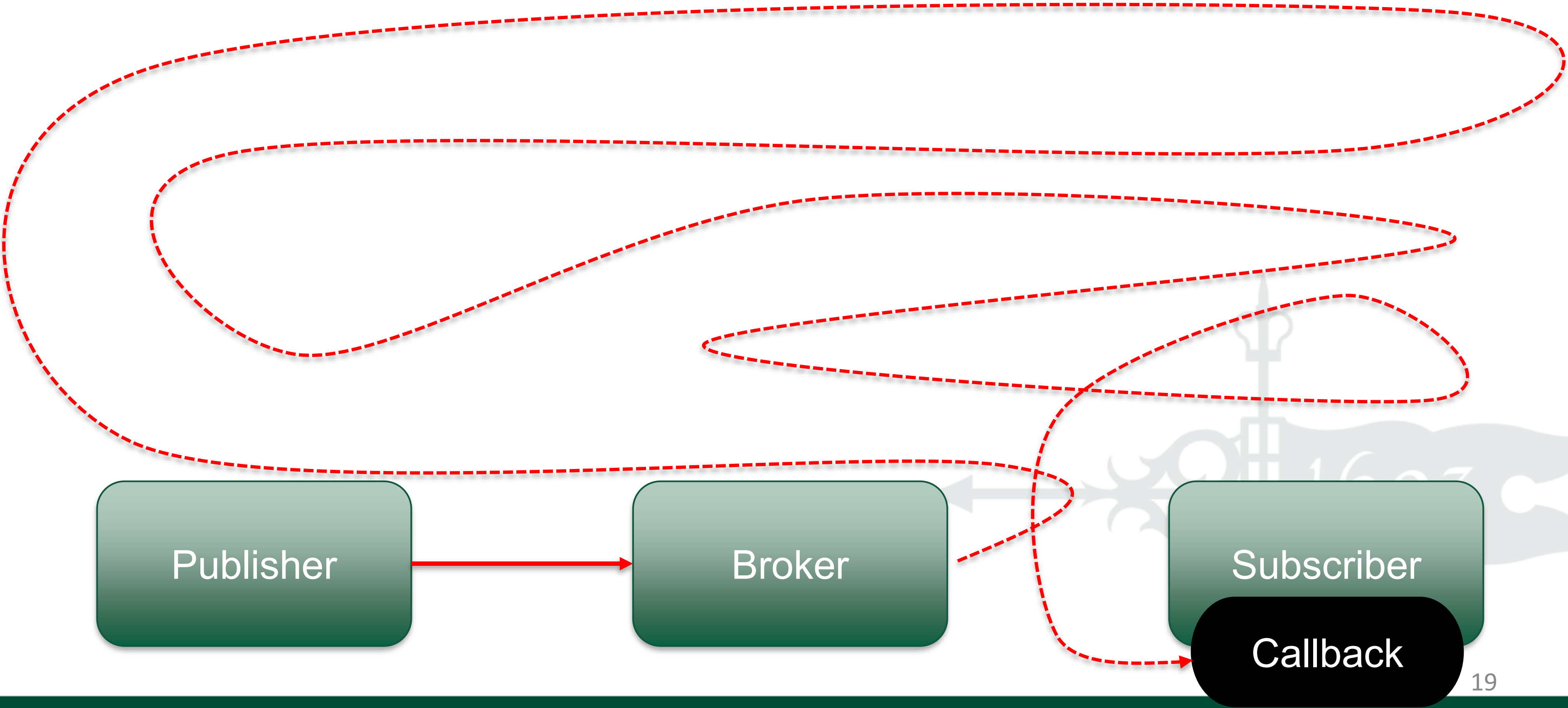
Pub/Sub: Time Decoupling



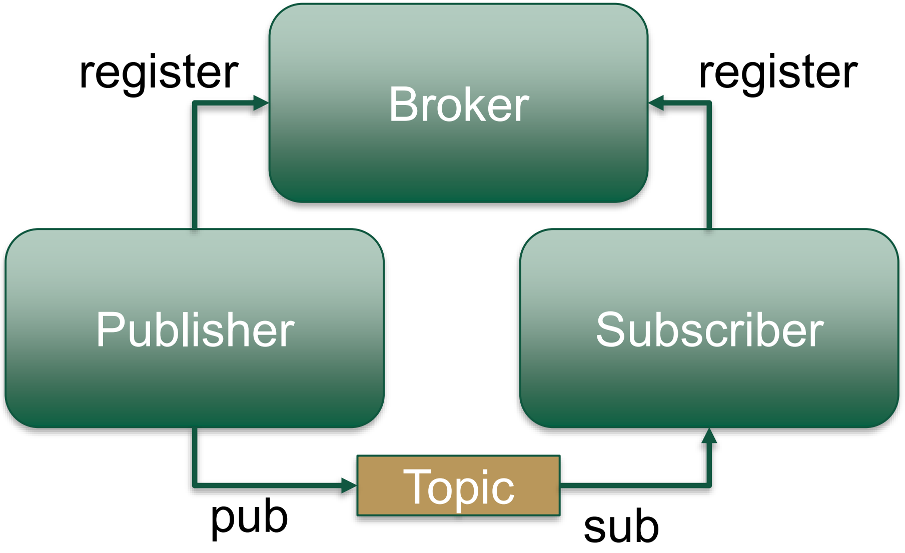
Pub/Sub: Time Decoupling



Pub/Sub: Time Decoupling

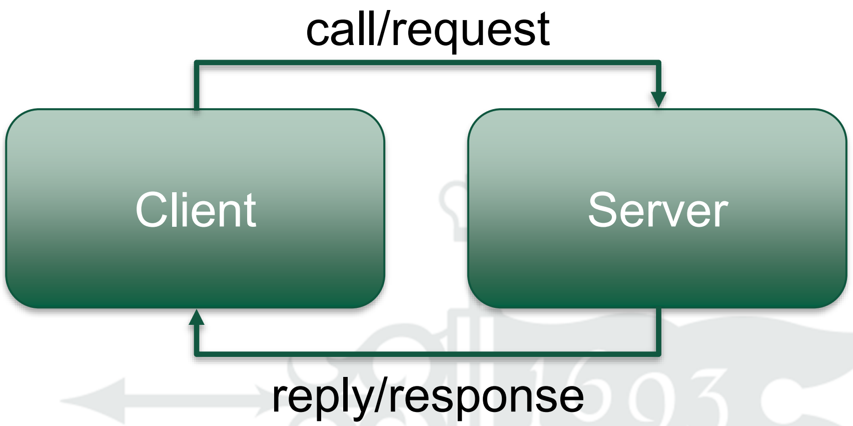


Pub/Sub vs Client Server



Asynchronous

Synchronous



ROS also allows **non-blocking** synchronous calls

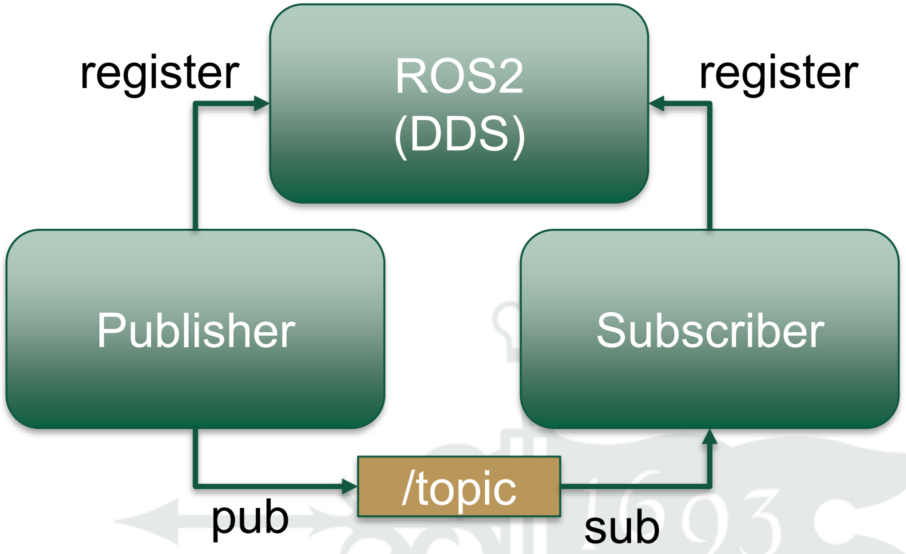
Pub/Sub Functionality

- Filtering
  - Topic-based
  - Content-based
- Message Delivery
  - How to physically transmit the message
  - Unicast, Multicast, push/pull

Pub/Sub in ROS2

- ROS2 connects publishers and subscribers
  - Uses Data Distribution Service (DDS) middleware

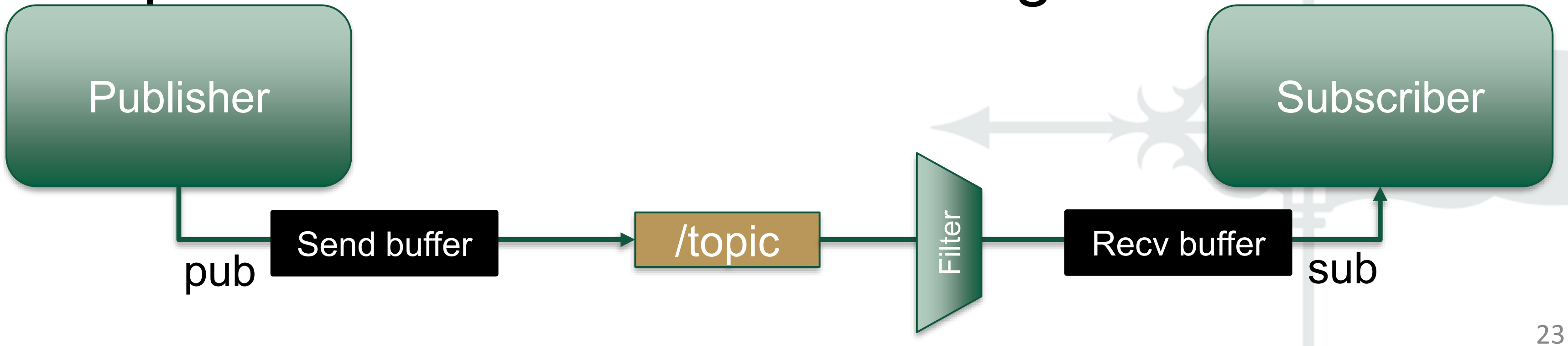
- Nodes can be publishers and/or subscribers to unlimited topics



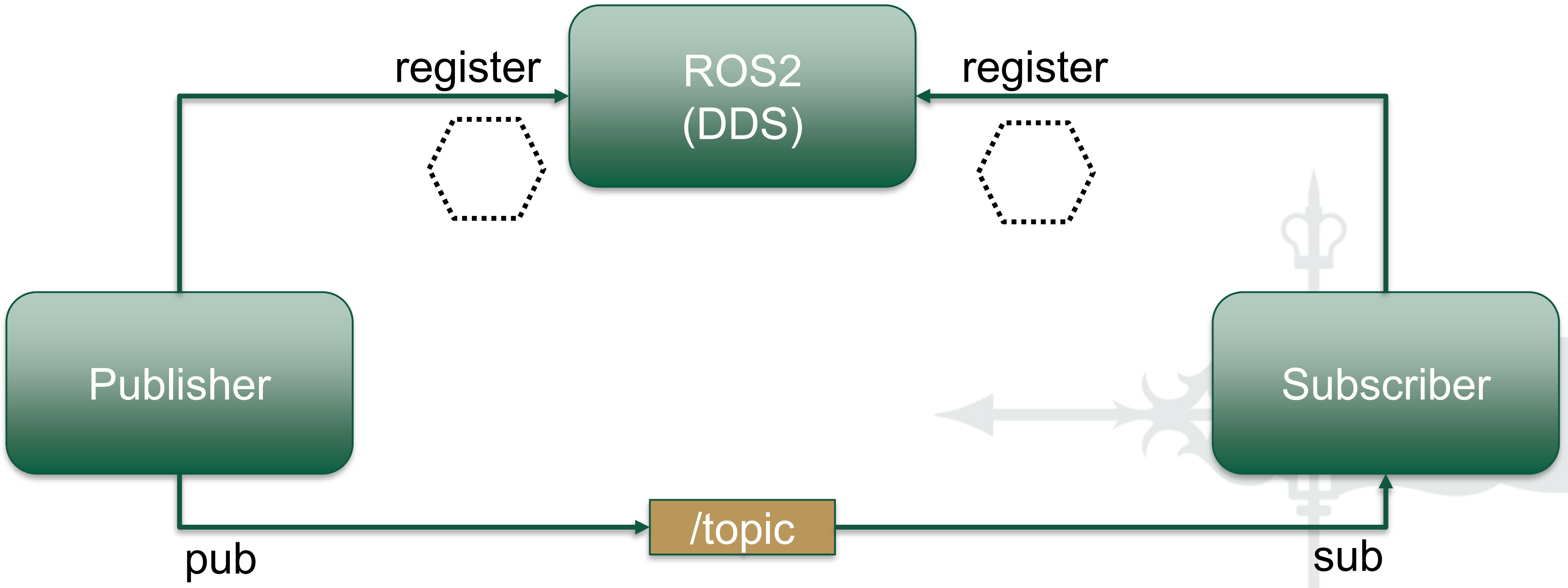
ROS also allows client-server and action paradigms

ROS2 Topic Communication

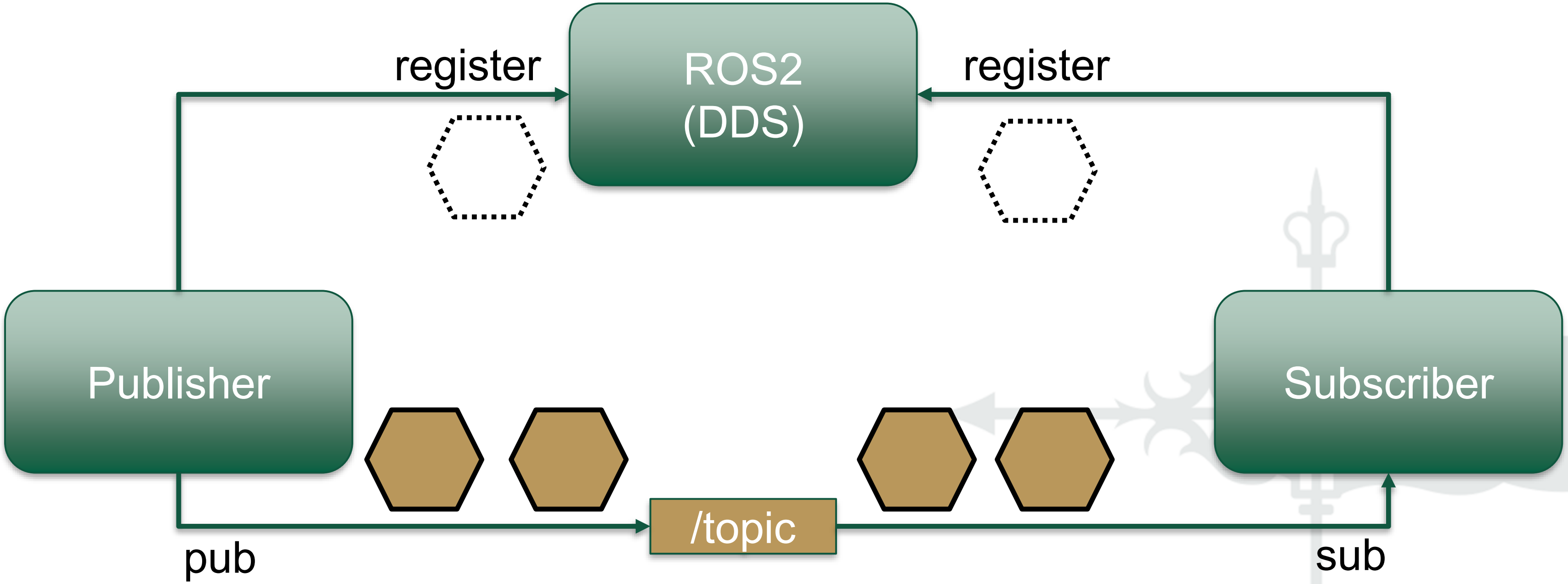
- All pub/sub uses topic-based filtering
- Buffering
- Peer-to-peer
- Optional: content-based filtering



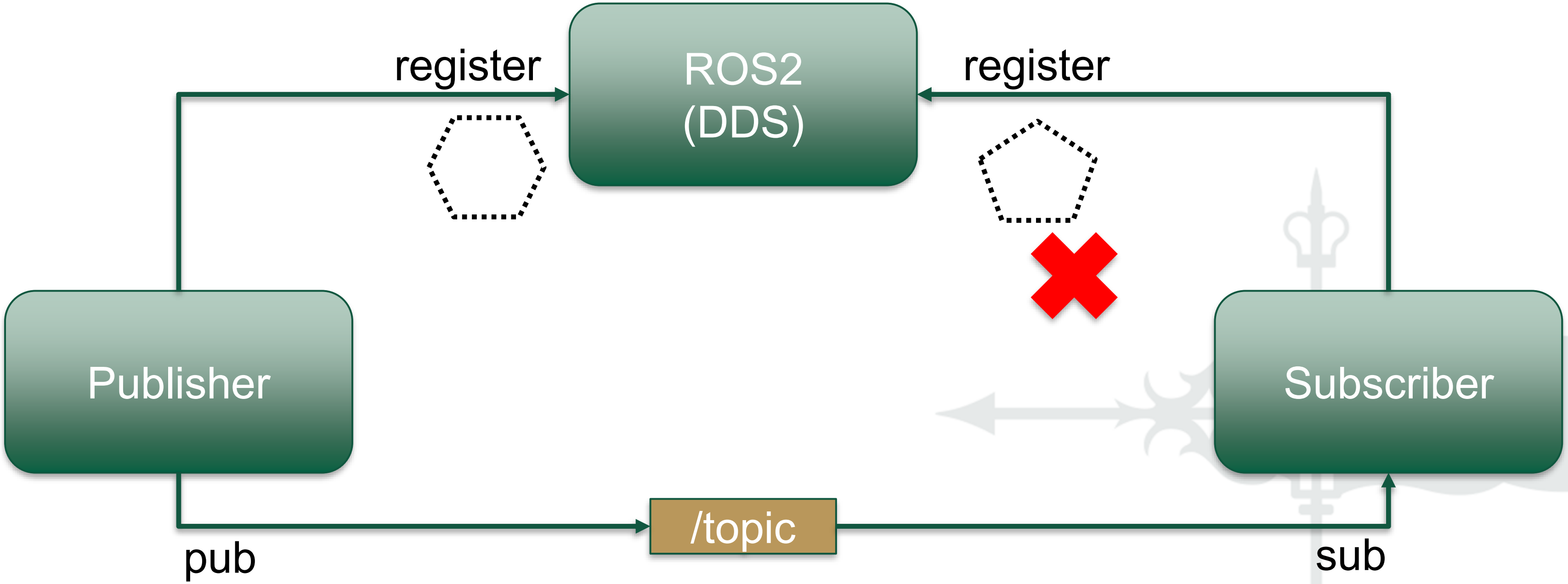
ROS2 Topic Message Types



ROS2 Topic Message Types



ROS2 Topic Message Types



ROS2 Topic Message Types

Geometry Msgs

Accel
AccelStamped
AccelWithCovariance
AccelWithCovarianceStamped
Inertia
InertiaStamped

Point

Point32
PointStamped
Polygon
PolygonInstance
PolygonInstanceStamped
PolygonStamped
Pose
Pose2D
PoseArray

PoseStamped

PoseWithCovariance
PoseWithCovarianceStamped
Quaternion
QuaternionStamped
Transform
TransformStamped
Twist
TwistStamped
TwistWithCovariance
TwistWithCovarianceStamped

Vector3

Vector3Stamped
VelocityStamped
Wrench
WrenchStamped

Standard Msgs

Bool

Byte
Char
Float32

Float64

Int8
Int16
Int32
Int64

String

UInt8
UInt16
UInt32
UInt64
ColorRGBA
Empty

Header

ByteMultiArray
Float32MultiArray
Float64MultiArray
Int8MultiArray
Int16MultiArray
Int32MultiArray
Int64MultiArray
MultiArrayDimension
MultiArrayLayout
UInt16MultiArray
UInt32MultiArray
UInt64MultiArray
UInt8MultiArray

Many Msg Packages

Sensor Msgs

- BatteryState
- Image
- LaserScan
- PointCloud
- Temperature

Simple Navigation

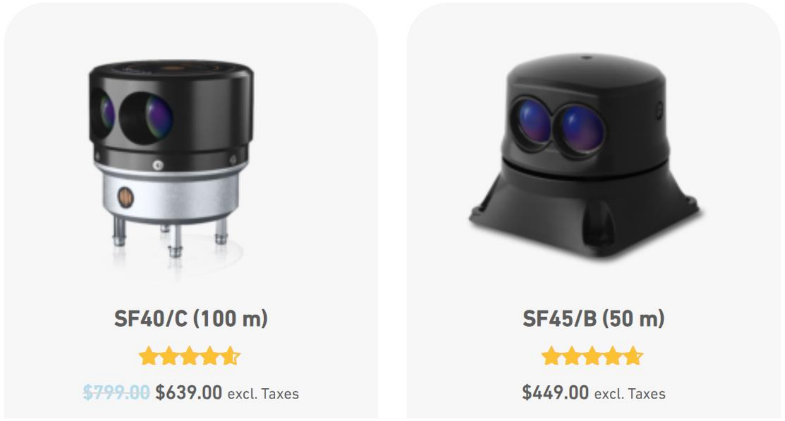
- OccupancyGrid

Advanced Navigation

- VoxelGrid

Support custom types

Pub/Sub Modularity

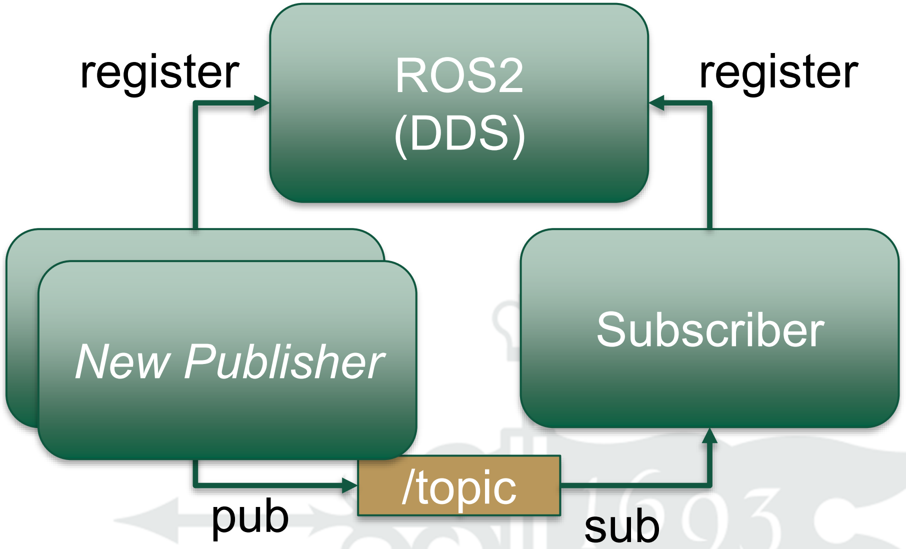

SF40/C (100 m)

~~$799.00~~ $639.00 excl. Taxes

SF45/B (50 m)

$449.00 excl. Taxes

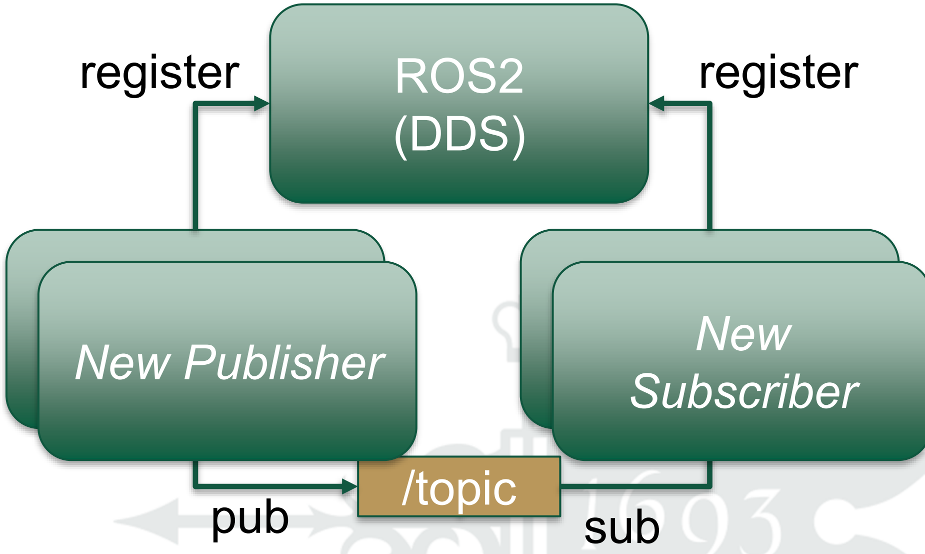
https://lightwarelidar.com/product-tag/scanning



Can replace pub/sub as new modules are available

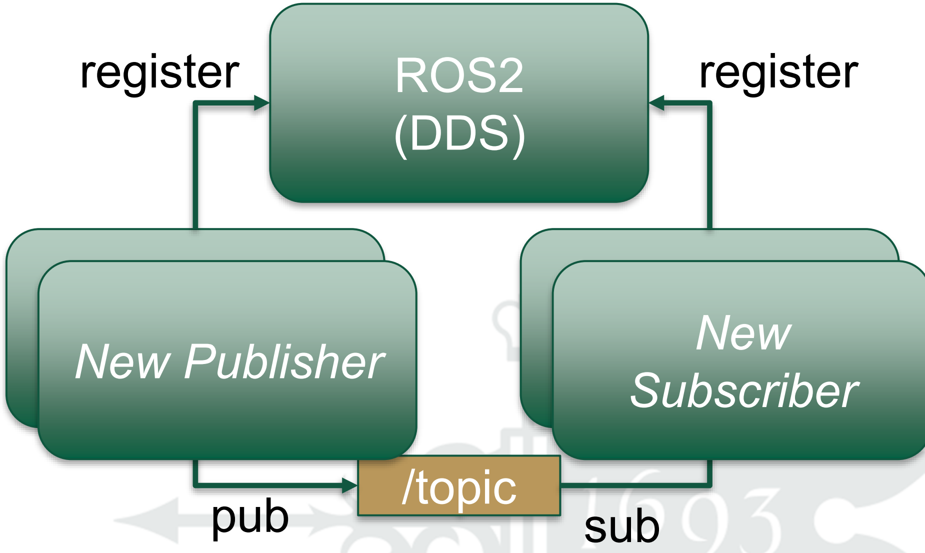
Leaky Abstractions

- What could leak?



Leaky Abstractions

- What could leak?
  - Improper Msg Use
  - No time guarantees
  - Lost messages
  - Buffer size matters?
- New Module:
  - Hyrum's Law

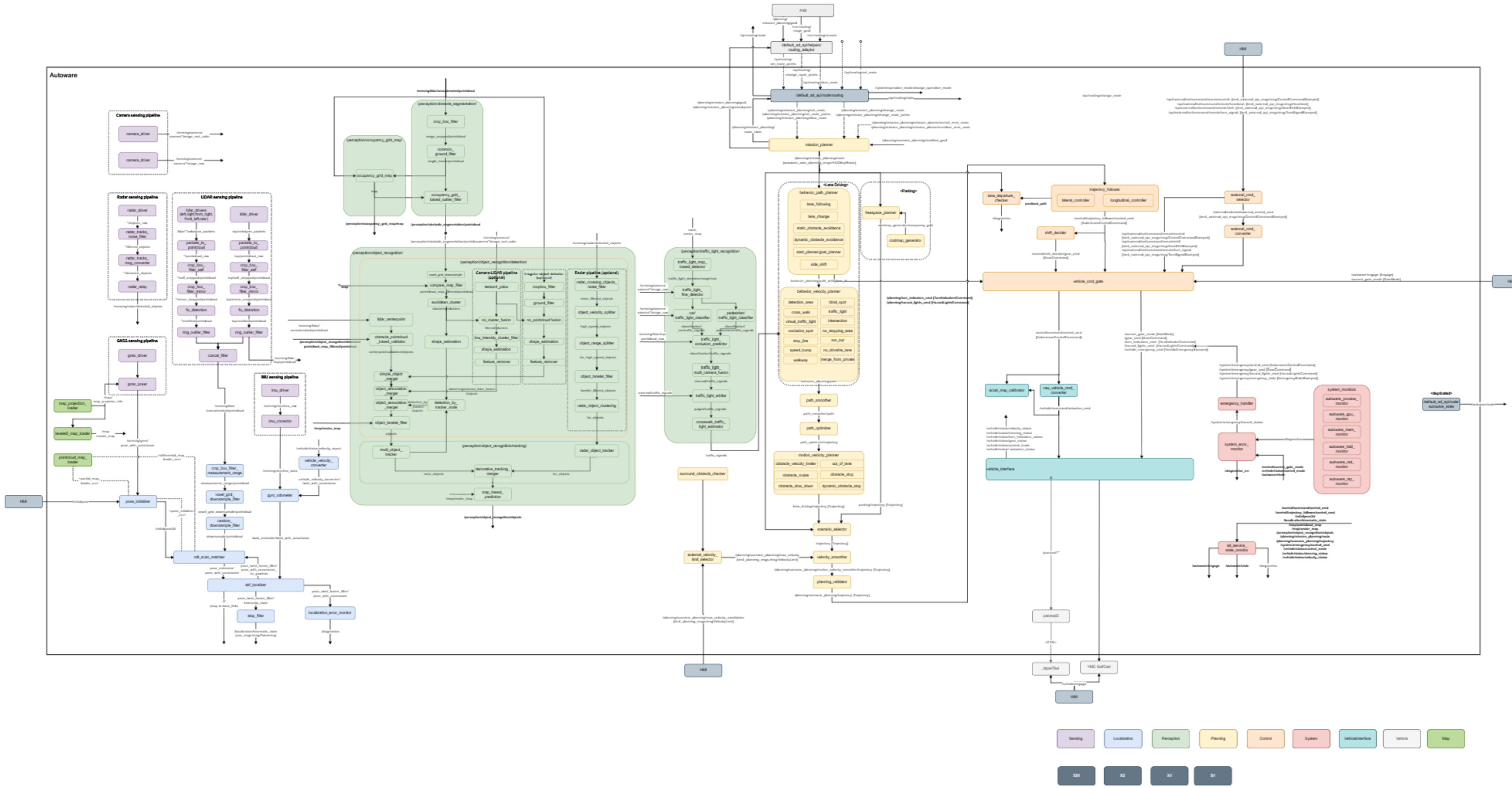


ROS2 – Real Systems!

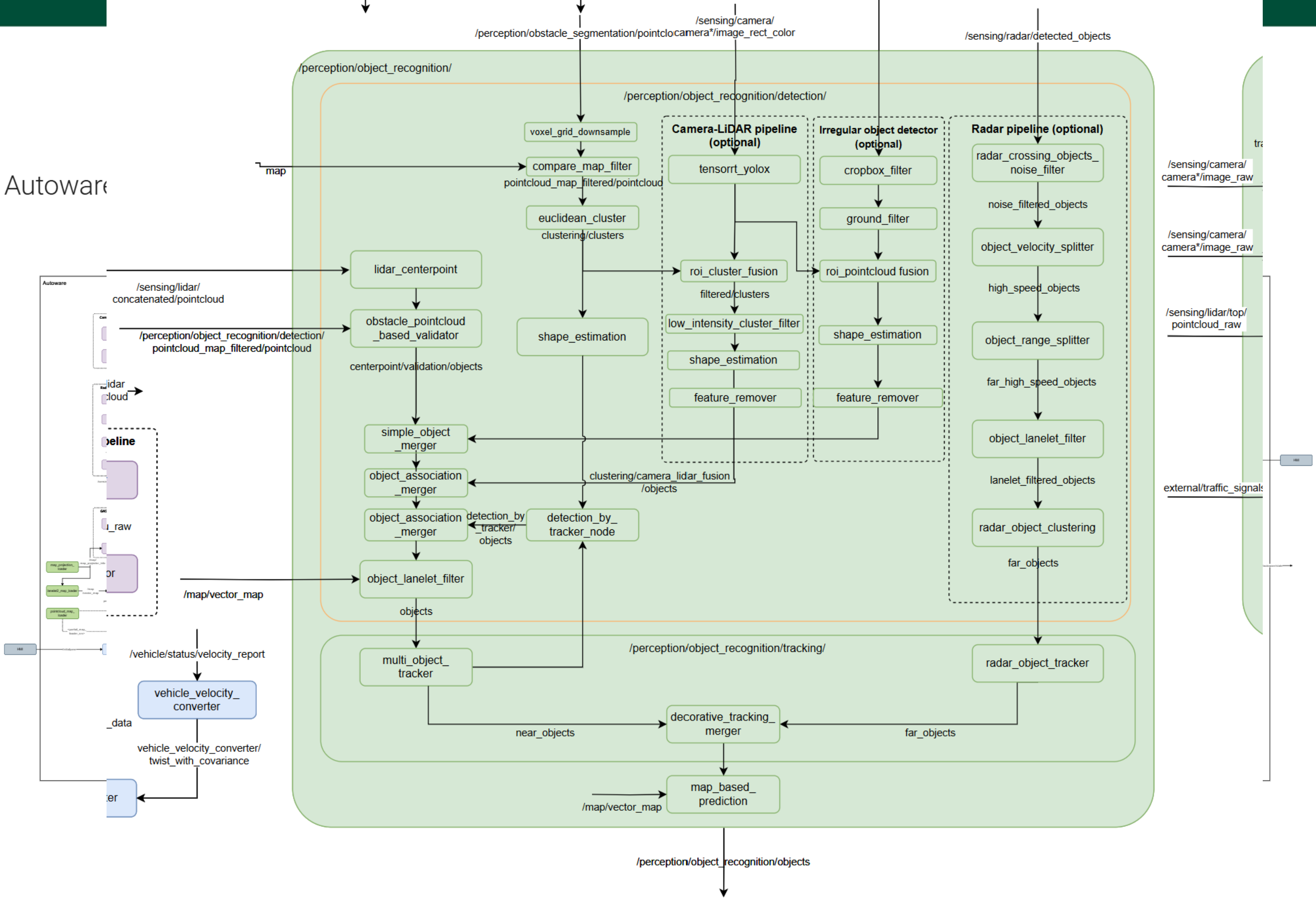
Autoware is the world’s leading open-source project dedicated to autonomous driving technology. Built on the Robot Operating System (ROS 2), Autoware facilitates the commercial deployment of autonomous vehicles across various platforms and applications.

ROS2 – Real Systems!

Autoware Universe



Autowar


/perception/obstacle_segmentation/pointcloud
/sensing/camera/camera*/image_rect_color
/sensing/radar/detected_objects
/perception/object_recognition/
/perception/object_recognition/detection/
voxel_grid_downsample
map
compare_map_filter
pointcloud_map_filtered/pointcloud
euclidean_cluster
clustering/clusters
Camera-LiDAR pipeline (optional)
tensorrt_yolox
roi_cluster_fusion
filtered/clusters
low_intensity_cluster_filter
shape_estimation
feature_remover
Irregular object detector (optional)
cropbox_filter
ground_filter
roi_pointcloud fusion
shape_estimation
feature_remover
Radar pipeline (optional)
radar_crossing_objects_noise_filter
noise_filtered_objects
object_velocity_splitter
high_speed_objects
object_range_splitter
far_high_speed_objects
object_lanelet_filter
lanelet_filtered_objects
radar_object_clustering
far_objects
/sensing/lidar/concatenated/pointcloud
lidar_centerpoint
/perception/object_recognition/detection/pointcloud_map_filtered/pointcloud
obstacle_pointcloud_based_validator
centerpoint/validation/objects
shape_estimation
simple_object_merger
object_association_merger
clustering/camera_lidar_fusion/objects
object_association_merger
detection_by_tracker/objects
detection_by_tracker_node
object_lanelet_filter
/map/vector_map
objects
/perception/object_recognition/tracking/
multi_object_tracker
radar_object_tracker
decorative_tracking_merger
near_objects
far_objects
map_based_prediction
/map/vector_map
/perception/object_recognition/objects
/vehicle/status/velocity_report
vehicle_velocity_converter
vehicle_velocity_converter/twist_with_covariance
_data
/sensing/camera/camera*/image_raw
/sensing/camera/camera*/image_raw
/sensing/lidar/top/pointcloud_raw
external/traffic_signals
Autoware
lidar cloud
peline
_raw
or
ter
tra

Core Robot SW Implications

- Development is complex and modular
  - Integration is key!
  - Abstractions are a must, but leaky!
- Asynchronous, event-driven, loosely coupled
  - Publish-Subscribe
  - Decentralized
  - Message Types