Core/Universe Localization Design Proposal

Doc Number :		
Created by:	*Reviewed by:*	*Approved by:*
Date	*Date*	*Date*

Purpose of this meeting

- Make an agreement about TierIV's architecture proposal and tf structure by discussing from both sides of AWF and TierIV

Current .Auto Localization Node Diagram (avp demo)

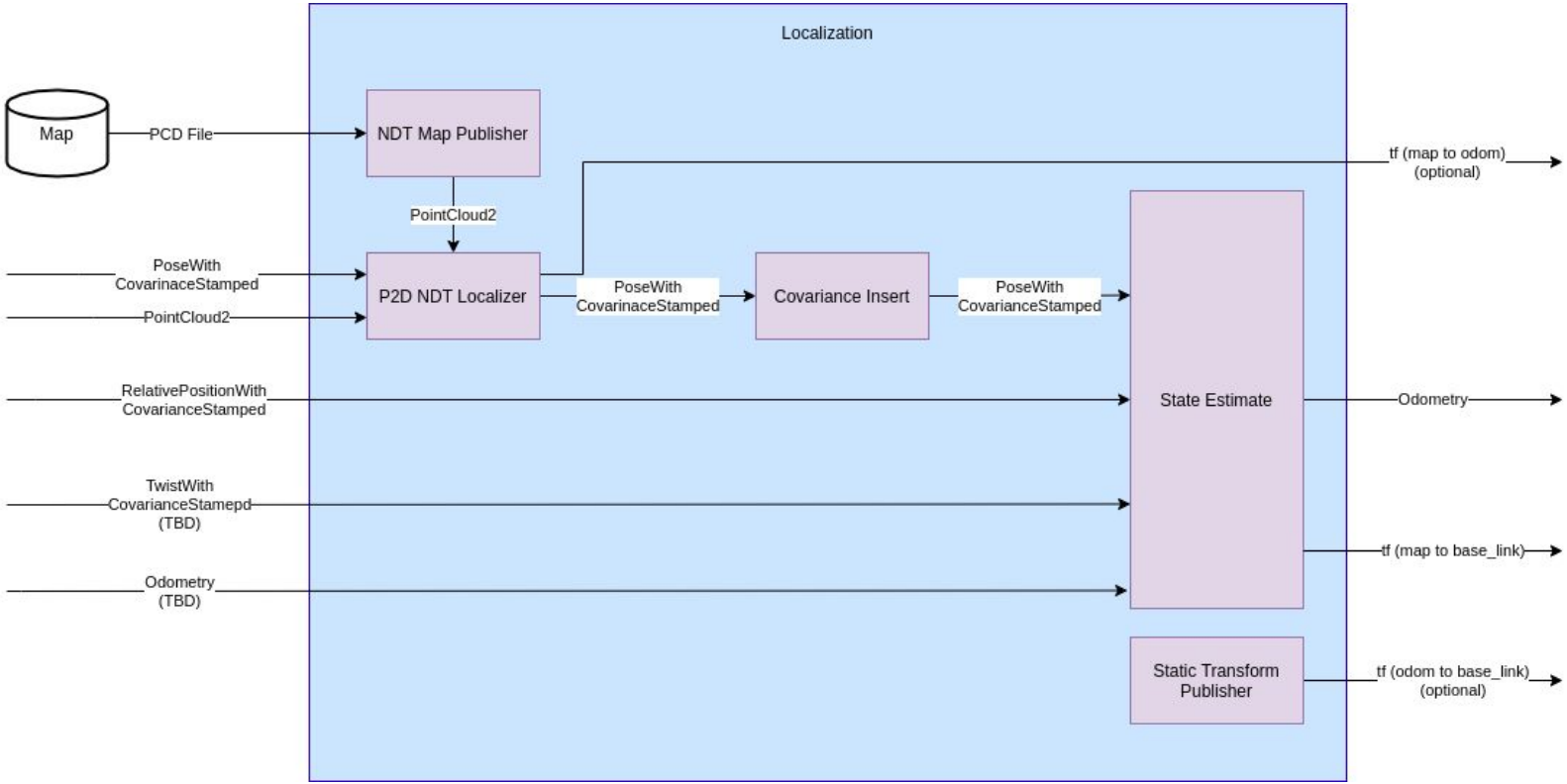


Figure: https://drive.google.com/file/d/1qv1SKN2zXzvW4RzQoCXSbCZP0P1Q5fbm/view?usp=sharing

.Auto architecture improvements

- Probably odom to base_link is not necessary
  - Since motion control will be performed on the map coordinates in our use-cases
- Need a monitor function to detect unexpected events (e.g. emergency stop)

Universe Node Diagram

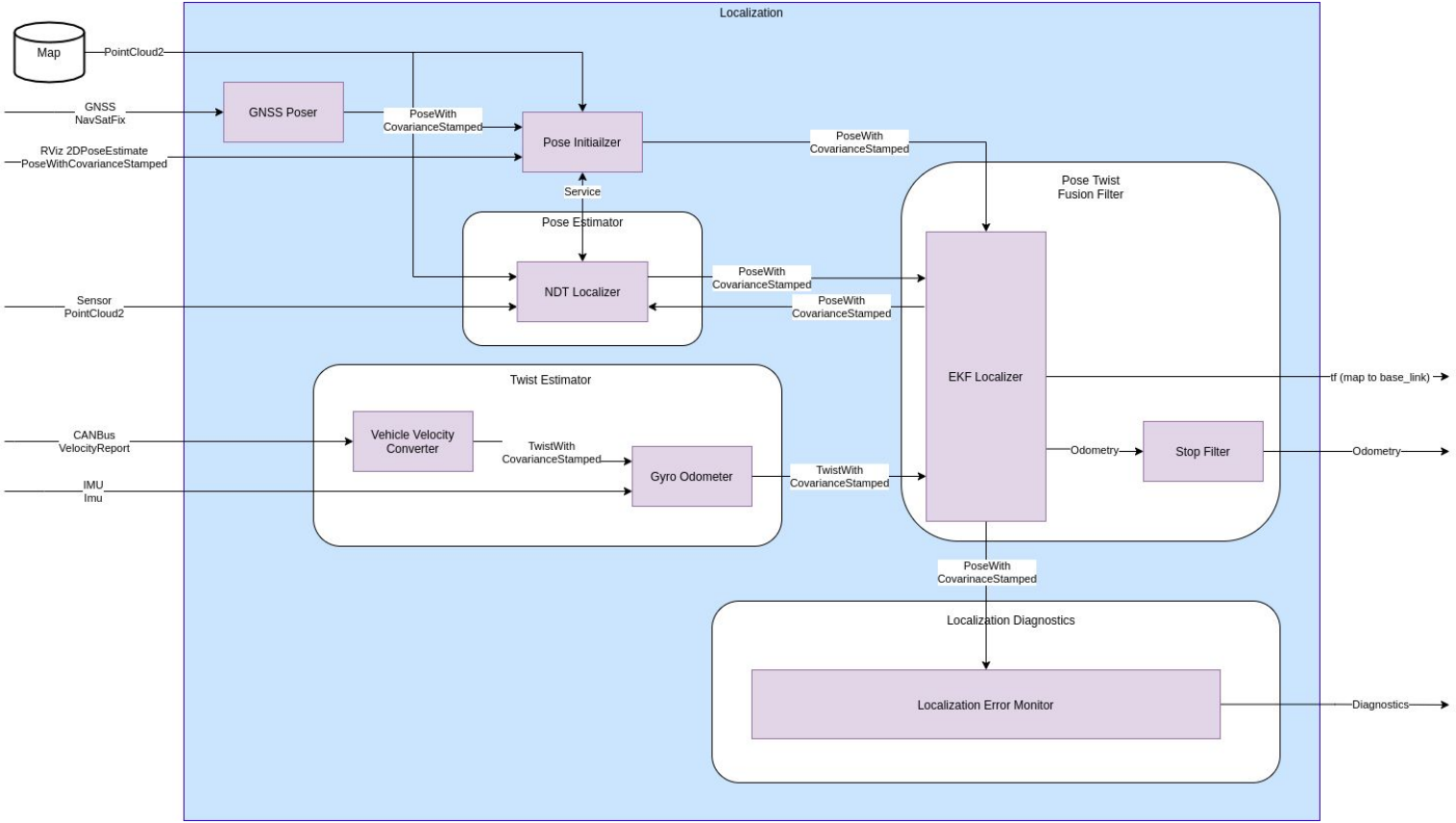


Figure: https://drive.google.com/file/d/1qv1SKN2zXzvW4RzQoCXSbCZP0P1Q5fbm/view?usp=sharing

To establish the architecture

- The role of each function should be defined according to how it is used (i.e. use-case)
- Modularize to improve reusability
- Improve flexibility to partially replace modules and functions
  - Kalman filter, Particle filter…
- Keep expandability and adaptability
  - To incorporate advanced functionalities (e.g. visual odometry)

[Design] Purpose and role of localization

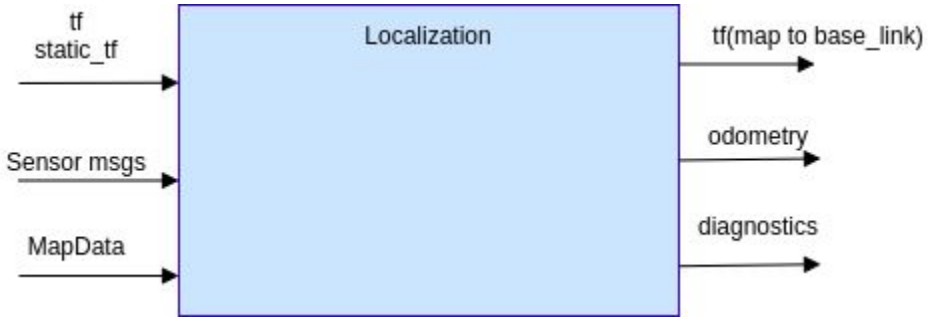
Role of localization

- Pose (translation, rotation) estimation on the map coordinates
- Twist (velocity, angular velocity) estimation on the ego vehicle coordinates

Use-case of localization in Autonomous Driving

- Planning
  - Path planning from the current vehicle location to the destination
  - Obtaining traffic rules from vector maps according to the current location
  - Obstacle avoidance
- Motion Control
  - Path following
  - Control for comfortable riding experience
- Perception
  - Recognizing surrounding objects (vehicles, pedestrians, traffic lights, etc.) on the map coordinates
- We don't take mapping into account

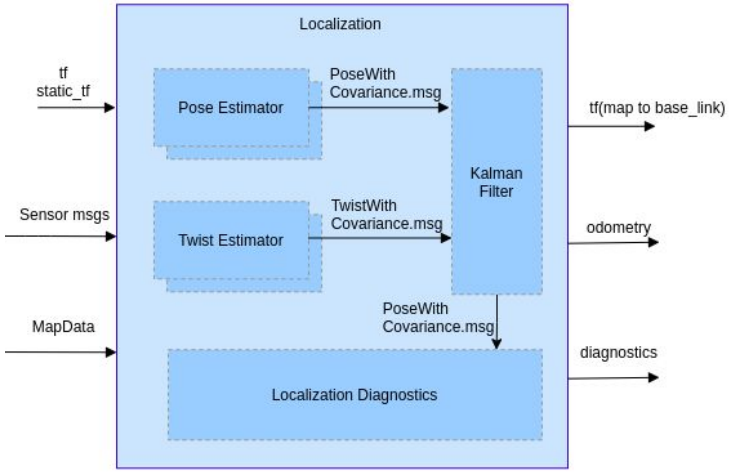
[Design] Architecture



- We just define input and output
  - odometery means
    pose (translation, rotation) in the map coordinates and
    twist (velocity, angular velocity) in the ego-vehicle coordinates
  - diagnostics is the signal for monitoring the localization state (warning, error, okay(ordinary))
- We don't set any restrictions for the internal structure
  - Instead we propose the recommended architecture
    since if people make modules as they like, the reusability will be reduced

Figure: https://drive.google.com/file/d/1qv1SKN2zXzvW4RzQoCXSbCZP0P1Q5fbm/view?usp=sharing

[Recommended] Architecture



We separately define the observation into pose and twist to improve reusability

- Pose estimator: estimates the pose (translation, rotation) on the map coordinates
- Twist estimator: estimates the twist (velocity, angular velocity) on the ego-vehicle coordinates

We are planning to develop .Core and .Universe based on this architecture

Figure: https://drive.google.com/file/d/1qv1SKN2zXzvW4RzQoCXSbCZP0P1Q5fbm/view?usp=sharing

Universe Node Diagram

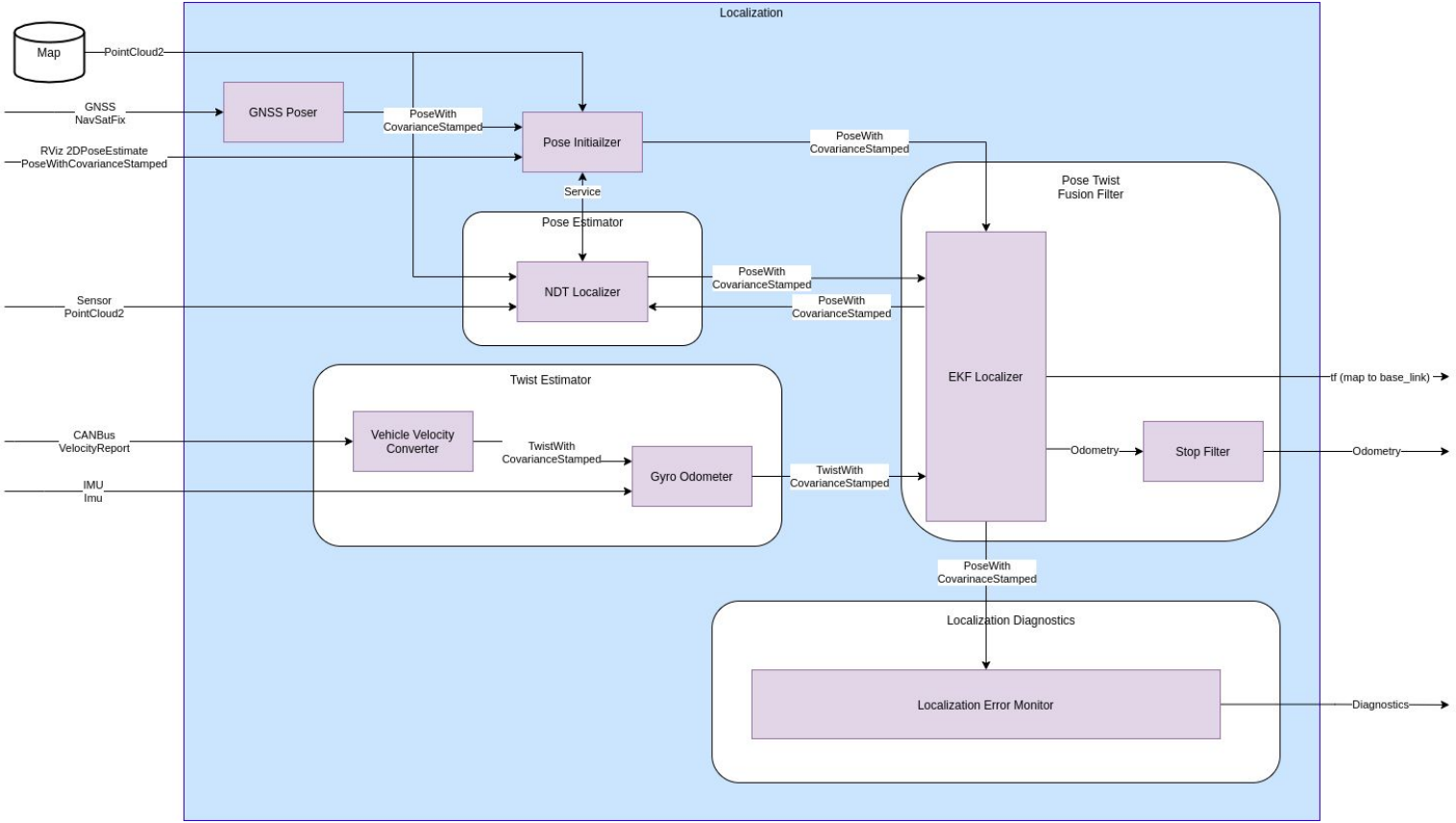


Figure: https://drive.google.com/file/d/1qv1SKN2zXzvW4RzQoCXSbCZP0P1Q5fbm/view?usp=sharing

[Design] TF Tree

earth : Origin of the earth coordinates (Earth Centered Earth Fixed)

map : Origin of the map coordinates (ex. MGRS-origin)

viewer: user-defined coordinate frame for rviz visualization

base_link: origin of the ego-vehice (the projection of rear-axle center)

odom and base_footprint can be added optionally

- But obviously they have to be managed appropriately

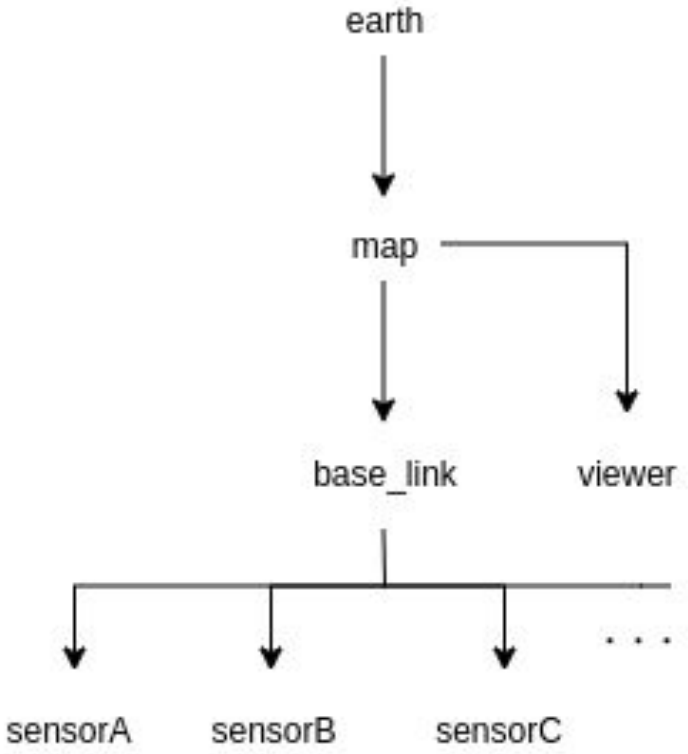


Figure: https://drive.google.com/file/d/1qv1SKN2zXzvW4RzQoCXSbCZP0P1Q5fbm/view?usp=sharing

Bus ODD Software Architecture

We take Bus ODD into account

- We designed our proposal architecture so that it can be utilized for Bus ODD

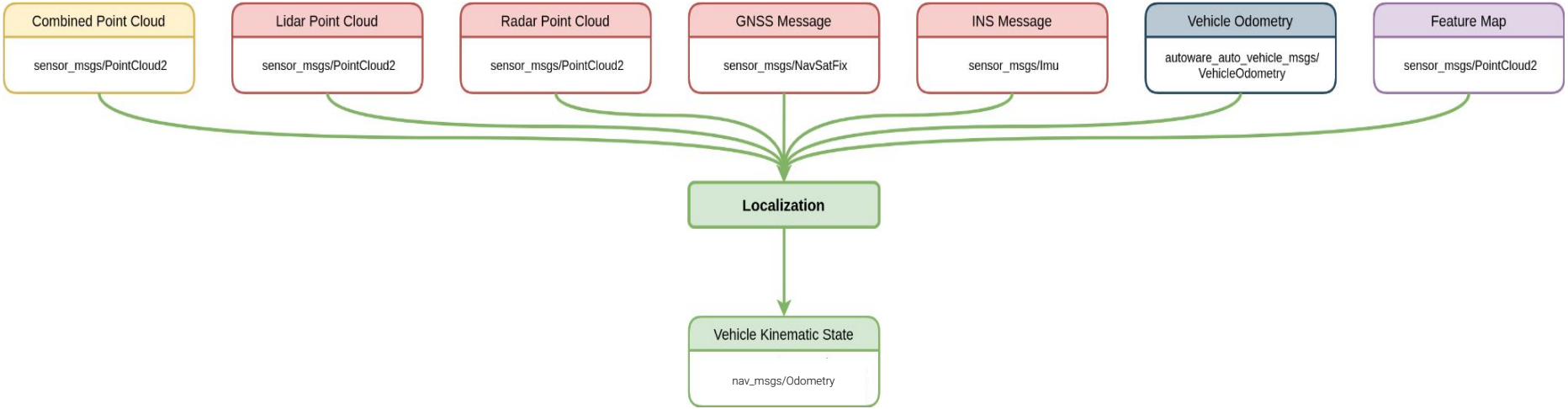


Figure: https://gitlab.com/autowarefoundation/autoware.auto/AutowareAuto/-/issues/1421