

# Balloon Popping Robair

Group 4

*Juan Vazquez, Charles Marshall, Sannara Ek, Shakeel Ahmad Sheikh*



# Our Project

## Robair:

- Intelligently identifies balloon targets while avoiding other targets, including human legs.
- Moves smoothly to the balloons
- Pops all balloons in the arena

\*We use an upside down laser for improved detection of balloons.

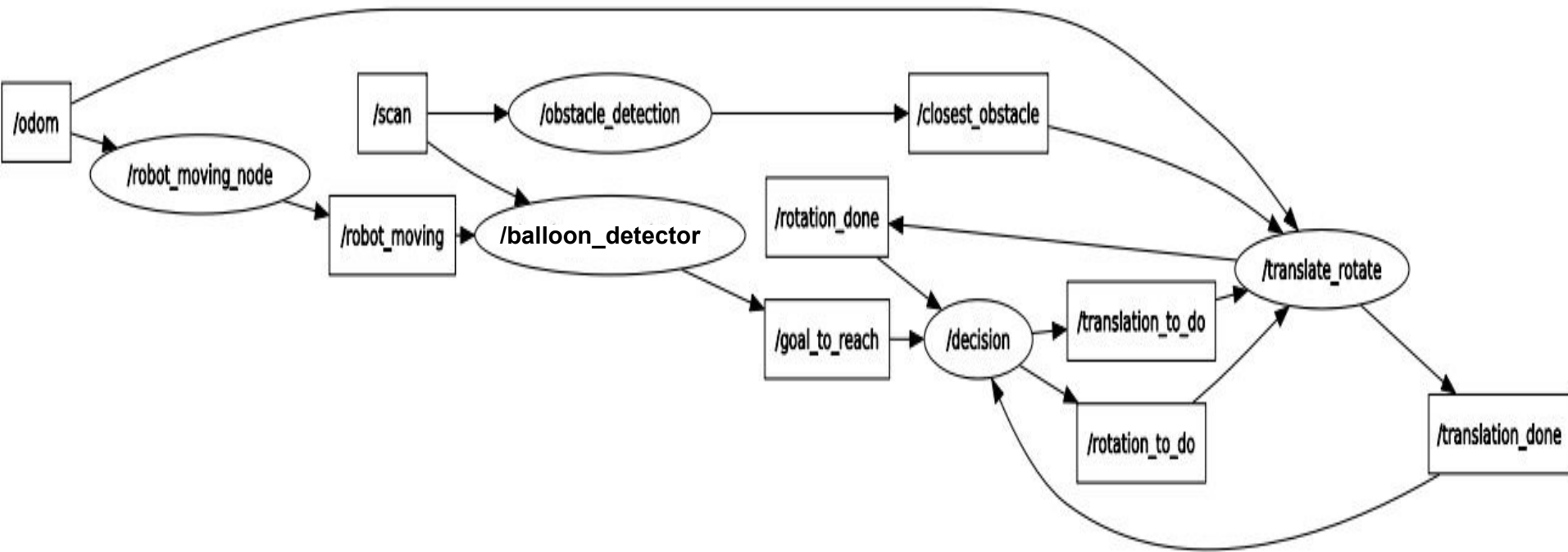
# Developments

- **Decision** → Decision\_Node.cpp
- **Vision** → Balloon\_detector\_node.cpp
- **Rotation / Translation** → Translate\_Rotate\_Node.cpp

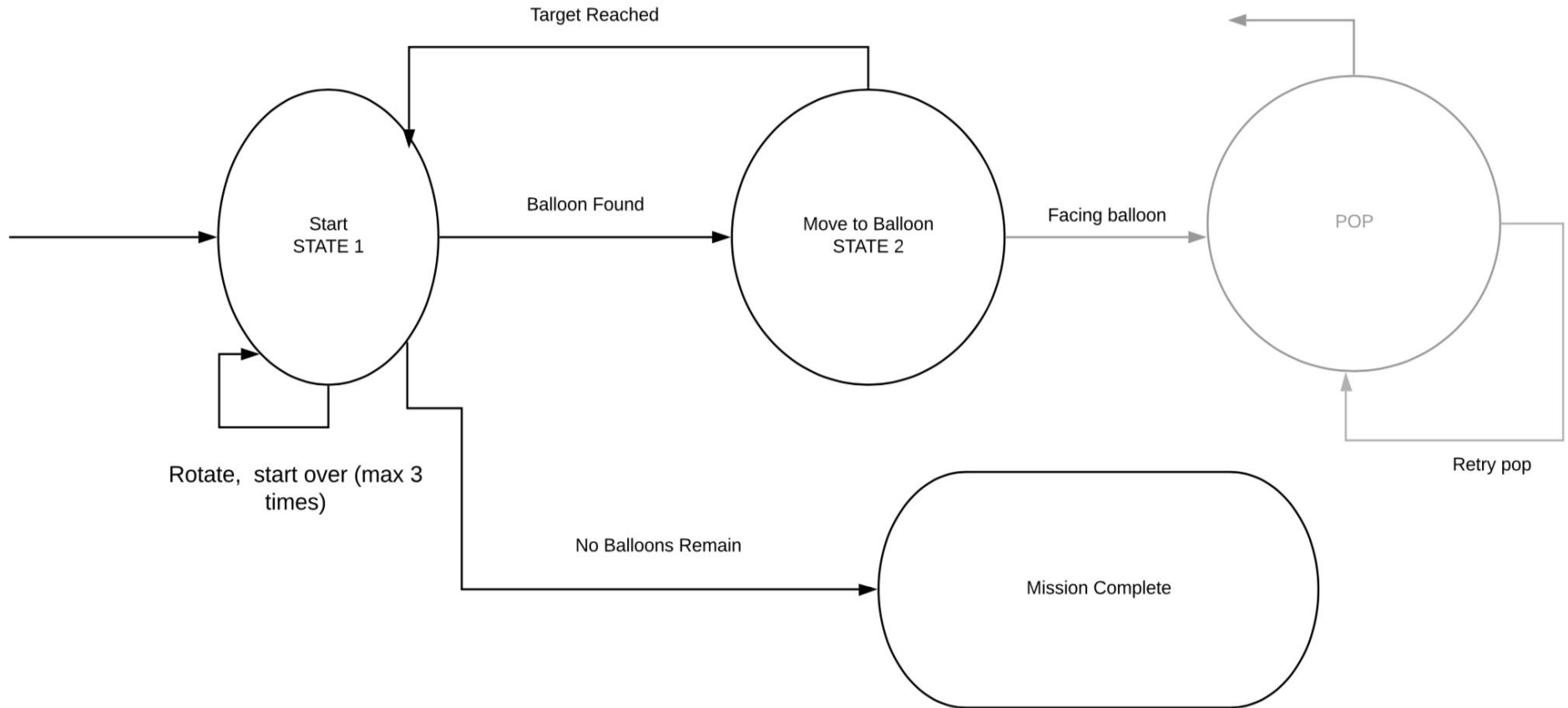
Additionally:

- Balloon Tracking → Balloon\_Tracker\_Node.cpp

# Architecture

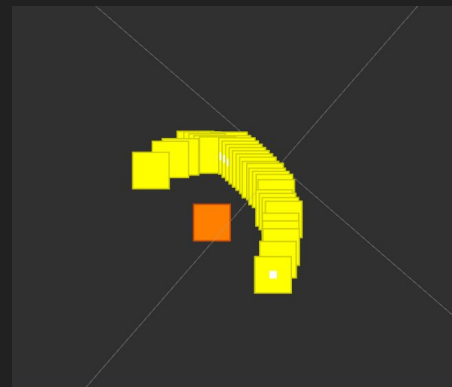


# Decision

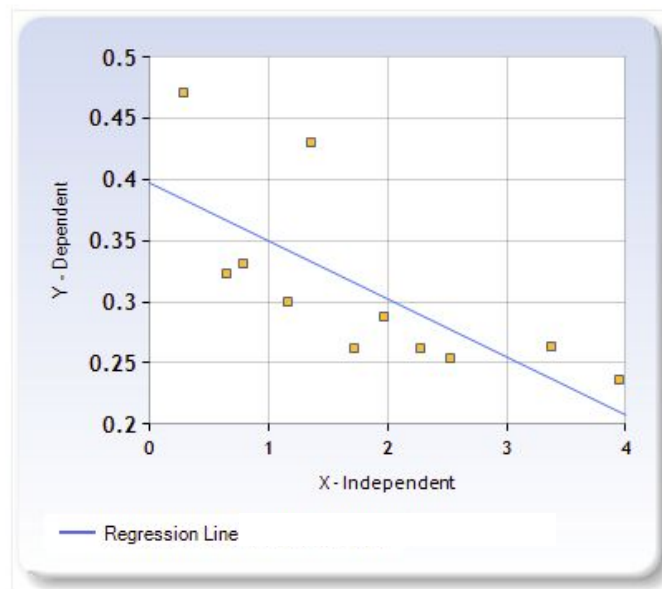


# Detecting the size of a balloon

- A Particular Width
- Collected Data
- A Linear Regression Problem
  - $BS = 0.05 * \text{Distance} - 0.28$
- Outliers to adjust

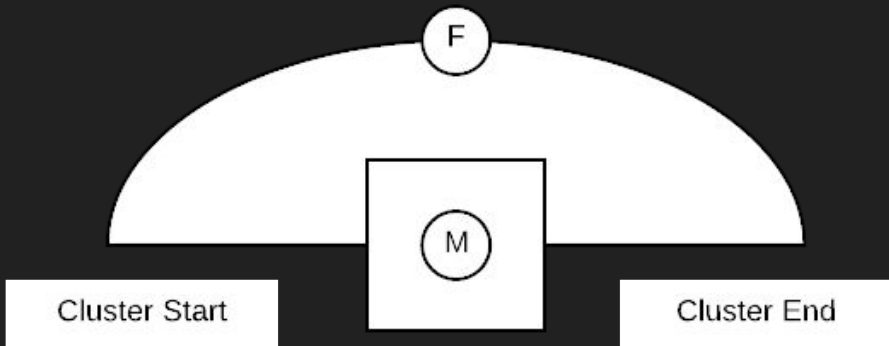
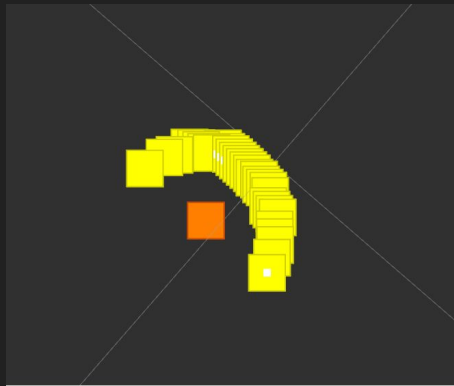


XValues	YValues
0.646	0.323
0.292	0.4708
1.72	0.2614
1.159	0.2997
2.28	0.262
3.369	0.264
3.94	0.236
2.524	0.254
1.966	0.288
0.796	0.331
1.355	0.43
M: 1.8225	M: 0.3109



# Detecting the curve of a balloon

- Calculate the middle(M) and the front point (F)
- Determine a threshold distance
- Calculate the distance between M and F
- Check if F is within the threshold



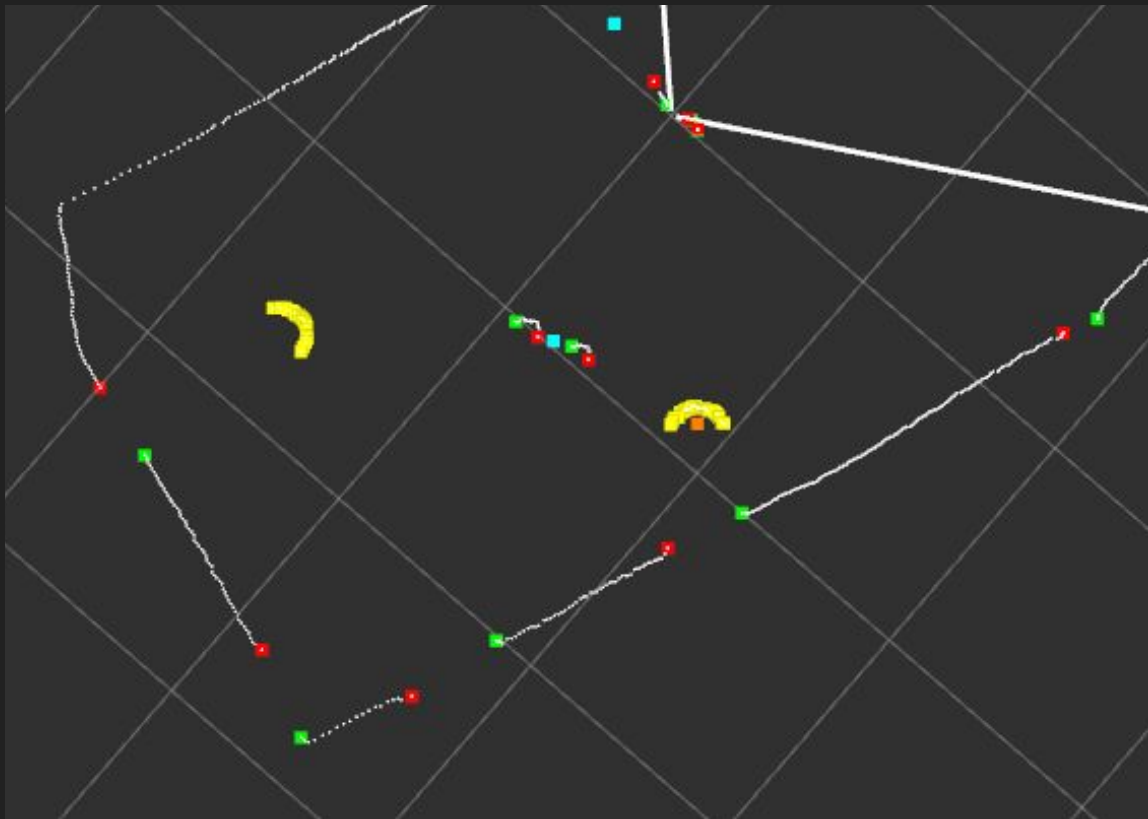


# Avoiding Legs

- Identify Human targets
- Filters out noises (Legs)
- Threshold for uncertainty

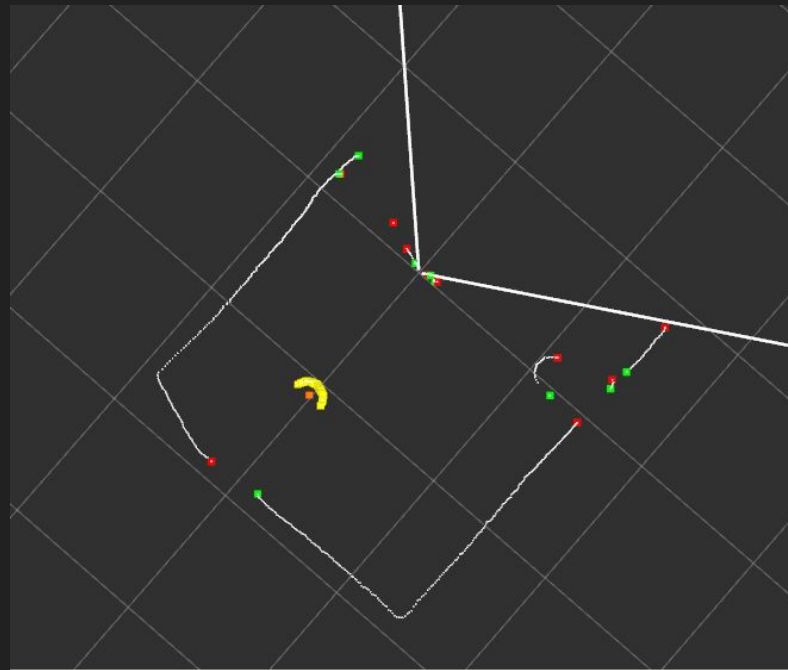
## Balloon Selections

- Closest Balloon



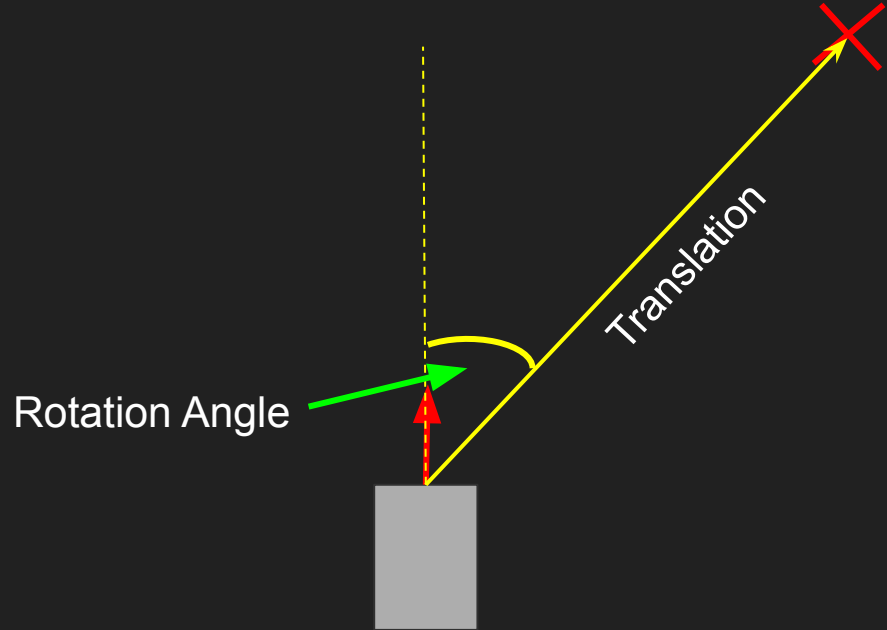
# Constraints with robair and balloons

- ❖ Using a upside down laser
- ❖ (some of the field of view is lost)
- ❖ Balloon Color Problem
- ❖ Possibility of using two lasers



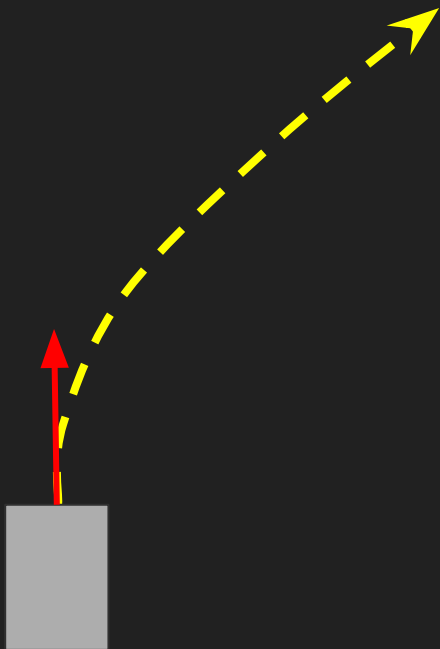
# Rotation / Translation

- Decision node:
  - Send rotation\_to\_do
  - Send translation\_to\_do
- Rotation/Translation:
  - Starts when both values have been received



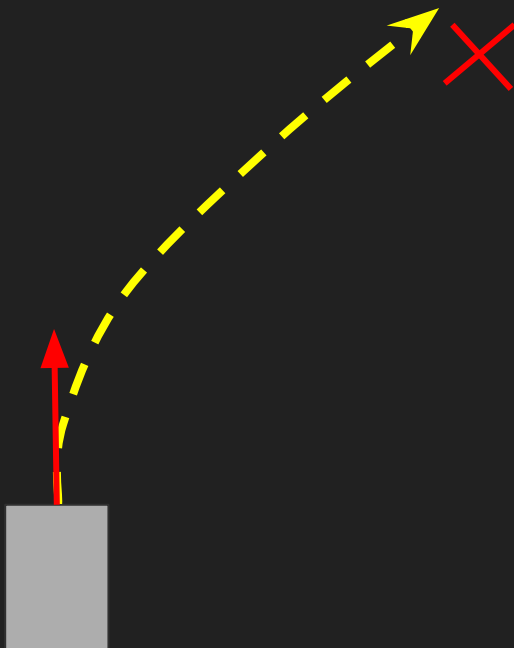
# Translate\_rotate node

- Waits for both rotation\_to\_do and translation\_to\_do
- Using PID, calculates rotation\_speed and translation\_speed
  - $W = 1 - \frac{\text{rotation\_speed}}{\text{THETA\_MAX}}$
  - Final translation speed
    - $\text{Translation\_speed} * w$

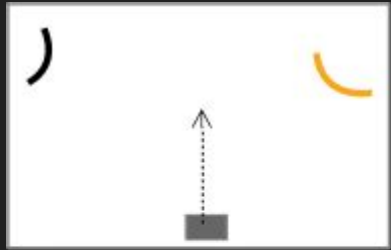


# Translate and rotate: getting to objective

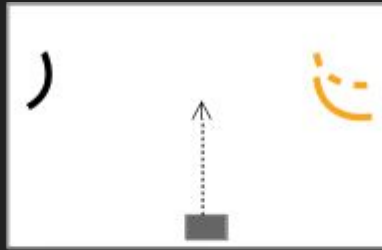
- We may miss the objective
  - Rely on the decision node logic
    - If objective is visible, go for it.
    - If not, rotate 90° until an objective is found



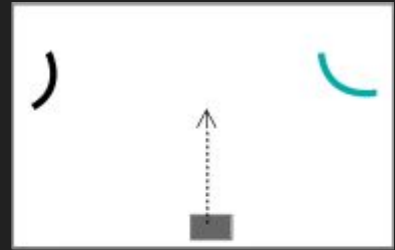
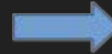
# Extra feature: Balloon tracking\*



- Perform balloon detection
- Save position of a balloon (Tracked balloon)



- Perform balloon detection
- Find balloon within threshold distance from tracked balloon



- If not balloon found, assume is still in previous position

\*Not fully integrated in the system

# Final Remarks

- Our work
- Our limitations
- Our ideas moving forward

# DEMO

(3 Parts)