

MIT 2.680
UNMANNED MARINE VEHICLE AUTONOMY,
SENSING, AND COMMUNICATIONS

Lecture 10 – Multi-Vehicle Mission Debugging

March 16th, 2023




Web: <http://oceanai.mit.edu/2.680>
Email:
Mike Benjamin, mikerb@mit.edu


MIT 2.680 Spring 2023 – Marine Autonomy – “Multi-Vehicle Missions”

Photo by Arjan Vermeij
GLINT '09

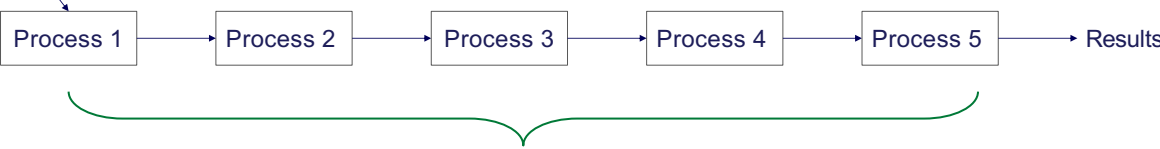
1



Know Your Pipeline



DEPLOY



```

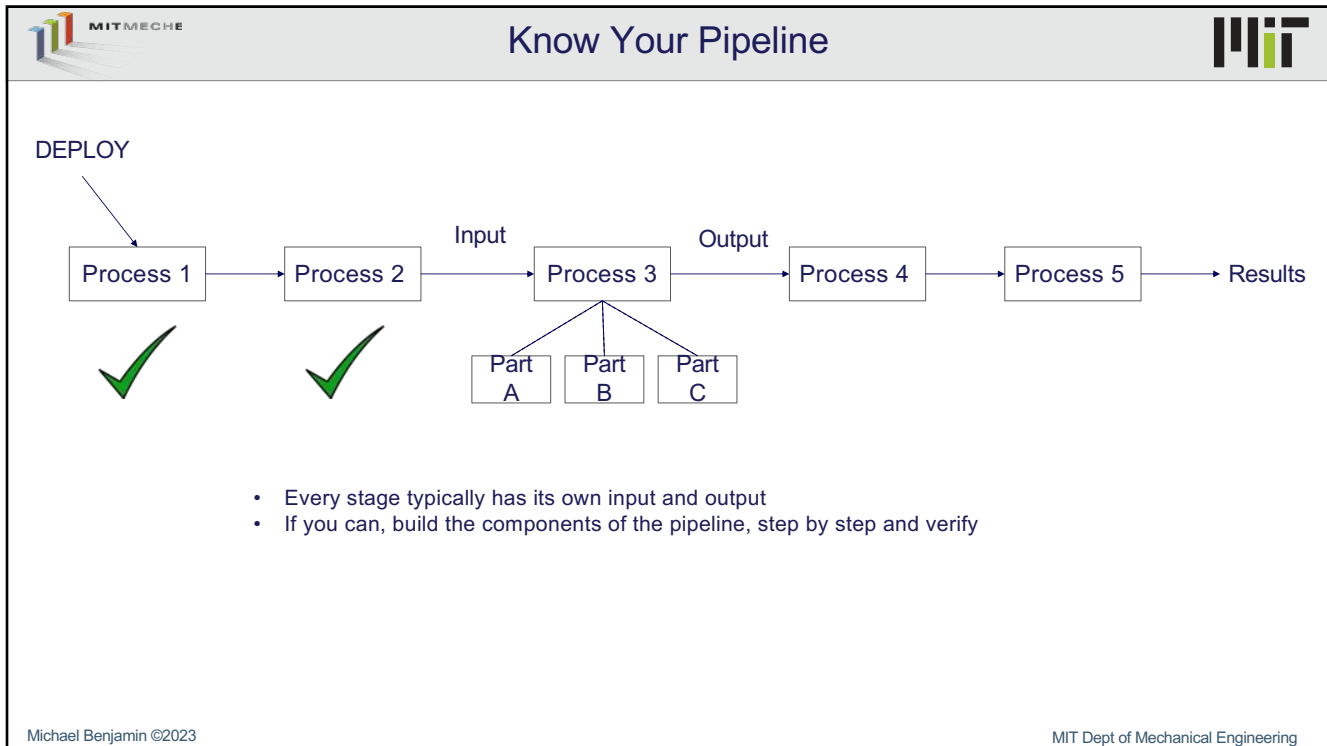
graph LR
    DEPLOY --> P1[Process 1]
    P1 --> P2[Process 2]
    P2 --> P3[Process 3]
    P3 --> P4[Process 4]
    P4 --> P5[Process 5]
    P5 --> Results
  
```

- A pipeline may involve several processes (e.g., MOOS Apps), handing a portion of the problem
- In development, and debugging, it is usually preferable to focus step by step.

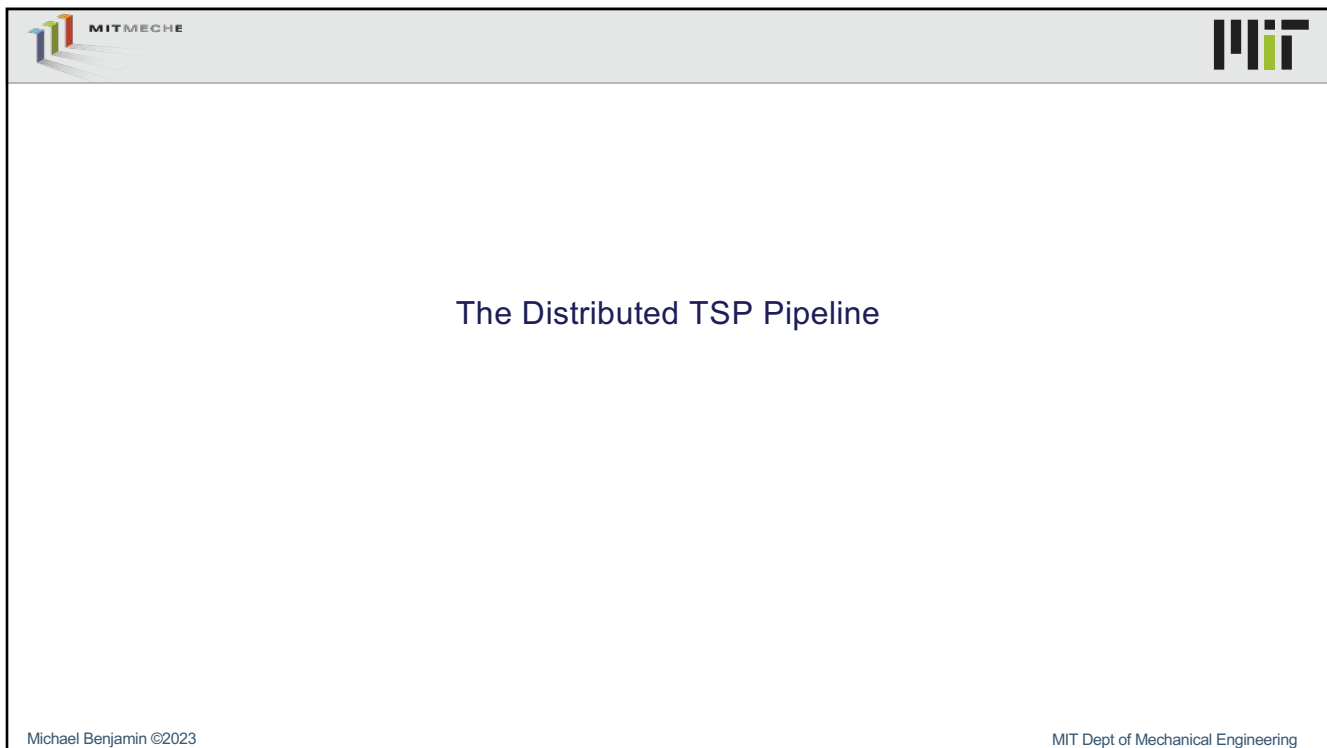
Michael Benjamin ©2023

MIT Dept of Mechanical Engineering

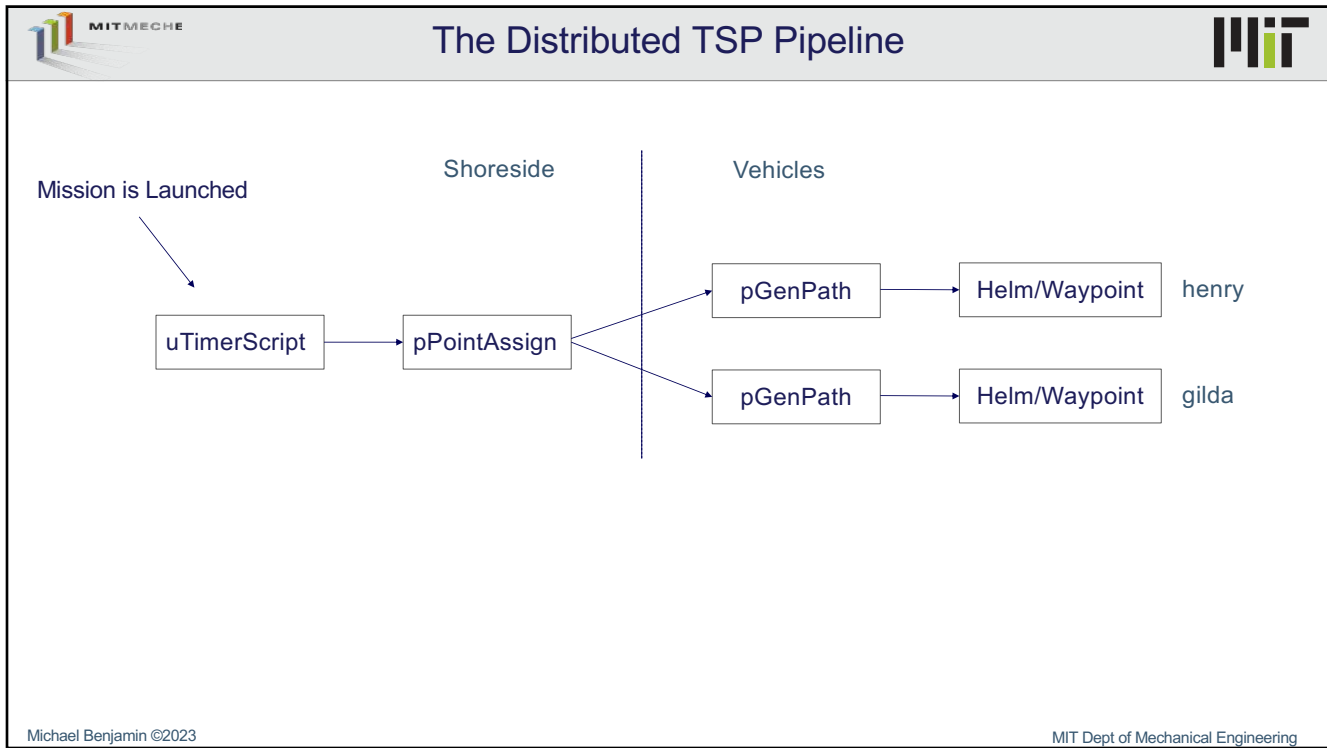
2



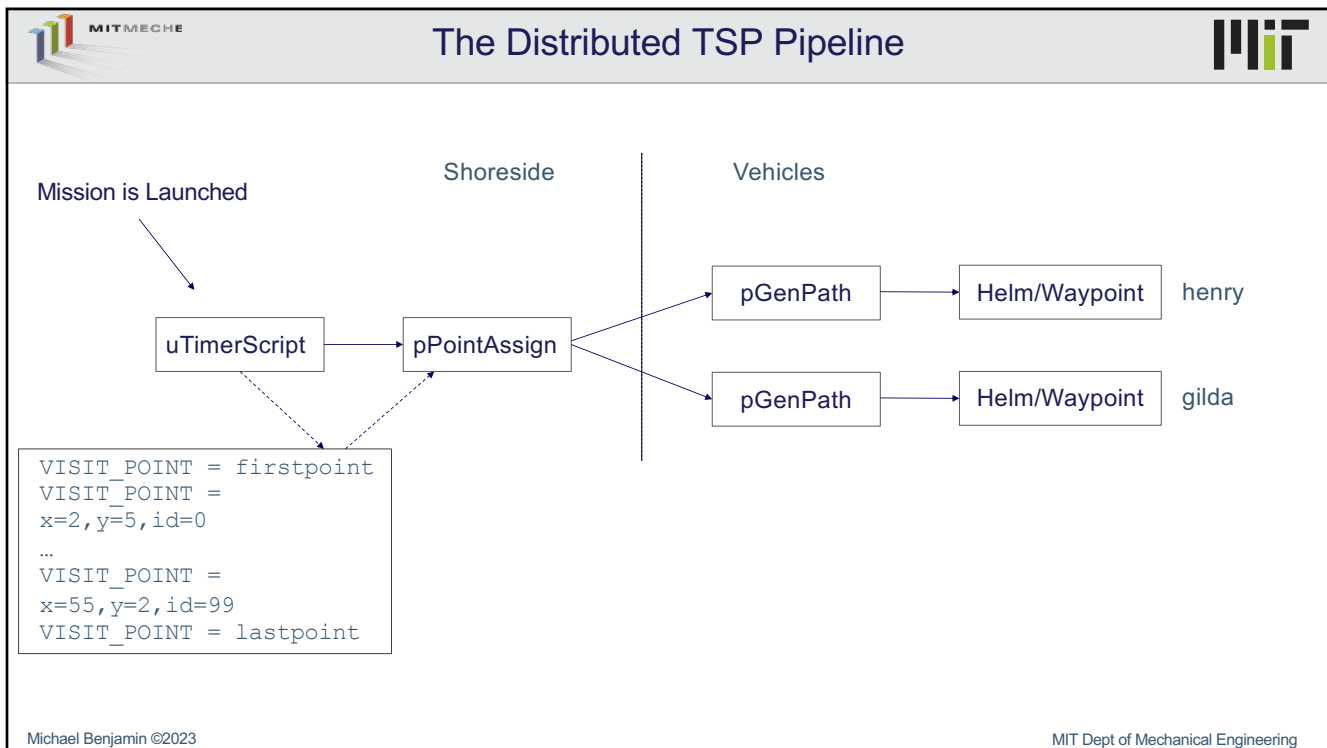
3



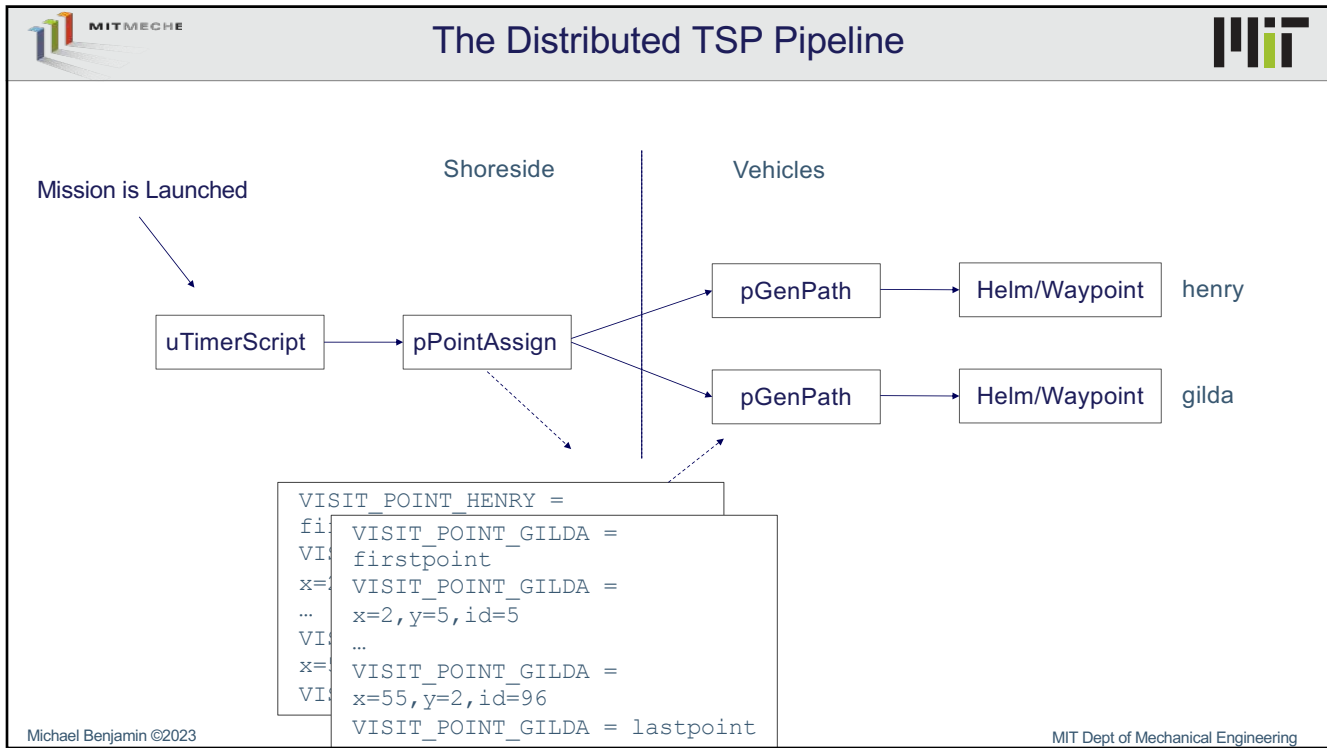
4



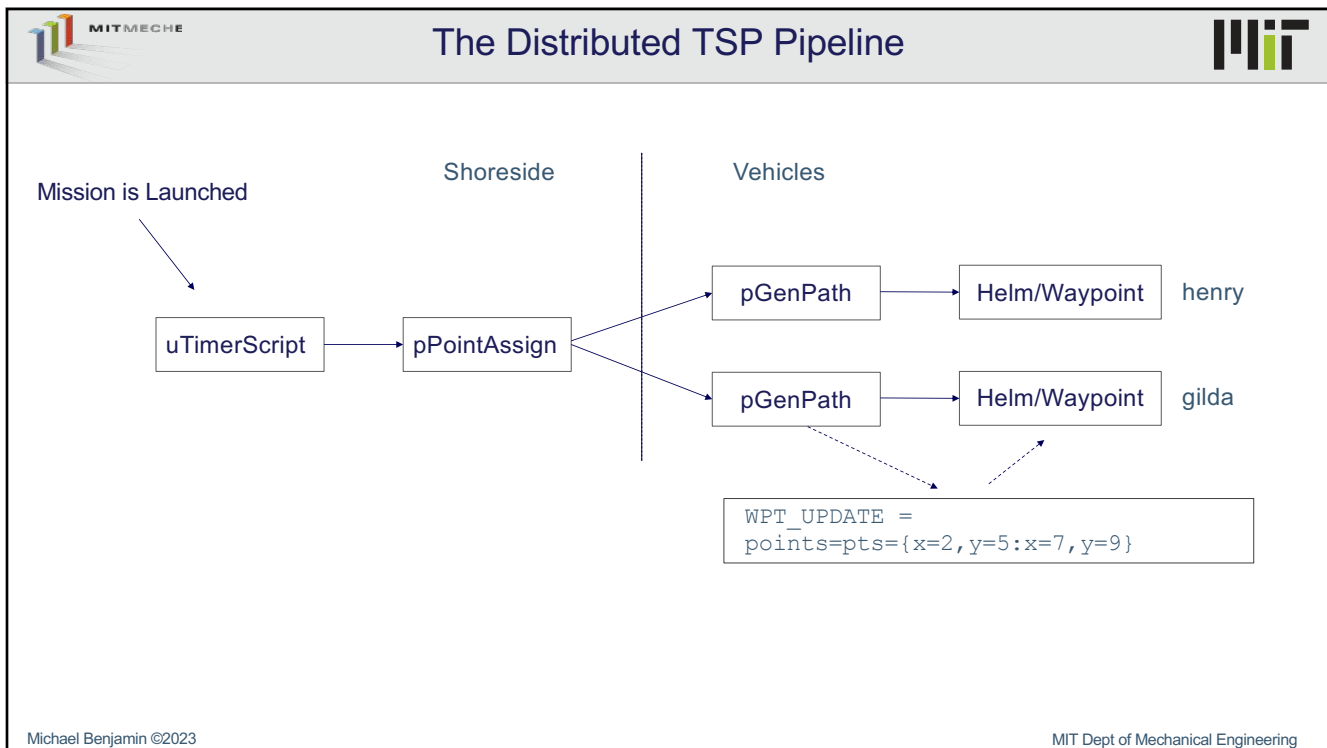
5





6



7



8





Implementing the Distributed TSP Pipeline


Michael Benjamin ©2023

MIT Dept of Mechanical Engineering

9



The Distributed TSP Pipeline Posting Points



Mission is Launched

↓

uTimerScript

Shoreside

→

pPointAssign

Vehicles

↗

pGenPath

→

Helm/Waypoint

henry

↘

pGenPath

→

Helm/Waypoint

gilda

```

VISIT_POINT = firstpoint
VISIT_POINT =
x=2,y=5,id=0
...
VISIT_POINT =
x=55,y=2,id=99
VISIT_POINT = lastpoint

```


- A timer script is added to the shoreside MOOS community
- See the uTimerScript documentation (Section 9.1) for a similar example
- Use the block_on parameter to ensure that pPointAssign is listening before posting

code
block_on = pPointAssign


Michael Benjamin ©2023

MIT Dept of Mechanical Engineering

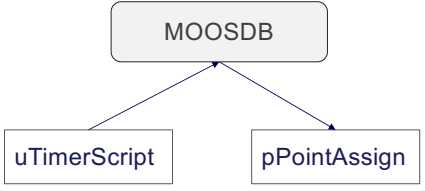
10



The Distributed TSP Pipeline Posting Points



What happens if uTimerScript starts well before pPointAssign?




```
shoreside.moos
Processconfig = ANTLER
{
  MSBetweenLaunches = 200

  Run = MOOSDB           @ NewConsole = false
  Run = uTimerScript      @ NewConsole = false
  Run = pShare            @ NewConsole = false
  Run = pMarineViewer     @ NewConsole = false
  Run = pLogger           @ NewConsole = false
  Run = pHostInfo         @ NewConsole = false
  Run = uFldShoreBroker   @ NewConsole = false
  Run = pPointAssign      @ NewConsole = false
}
```


Recall that when an app connects to the MOOSDB, it only gets the most recent mail for any registered variables.

Michael Benjamin ©2023
MIT Dept of Mechanical Engineering

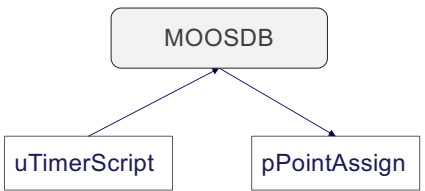
11



The Distributed TSP Pipeline Posting Points



We can “fix” this problem by moving uTimerScript to be the last app launched.




```
shoreside.moos
Processconfig = ANTLER
{
  MSBetweenLaunches = 200

  Run = MOOSDB           @ NewConsole = false
  Run = pPointAssign      @ NewConsole = false
  Run = pShare            @ NewConsole = false
  Run = pMarineViewer     @ NewConsole = false
  Run = pLogger           @ NewConsole = false
  Run = pHostInfo         @ NewConsole = false
  Run = uFldShoreBroker   @ NewConsole = false
  Run = uTimerScript      @ NewConsole = false
}
```


This is a brittle solution. Even if the person editing the .moos file were to put a comment in the file explaining why uTimerScript must be launched last.

Michael Benjamin ©2023
MIT Dept of Mechanical Engineering

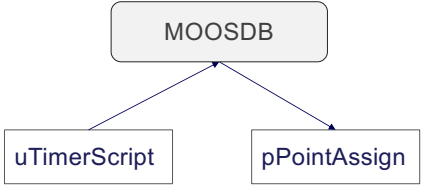
12



The Distributed TSP Pipeline Posting Points



Another arguably better fix is to delay the timer script for some number of seconds.



```
shoreside.moos
ProcessConfig = uTimerScript
{
  AppTick    = 3
  CommsTick  = 3

  paused     = false
  reset_time = all-posted
  reset_max  = 0
  delay_start = 10


  rand_var   = ...
  rand_var   = ...

  event = ...
}
```


This still feels brittle. What's the right amount of time?

Michael Benjamin ©2023
MIT Dept of Mechanical Engineering

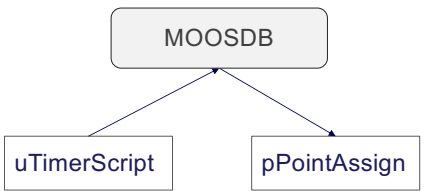
13



The Distributed TSP Pipeline Posting Points



A more durable fix is to ask uTimerScript to wait until pPointAssign is connected to the MOOSDB.



```
shoreside.moos
ProcessConfig = uTimerScript
{
  AppTick    = 3
  CommsTick  = 3

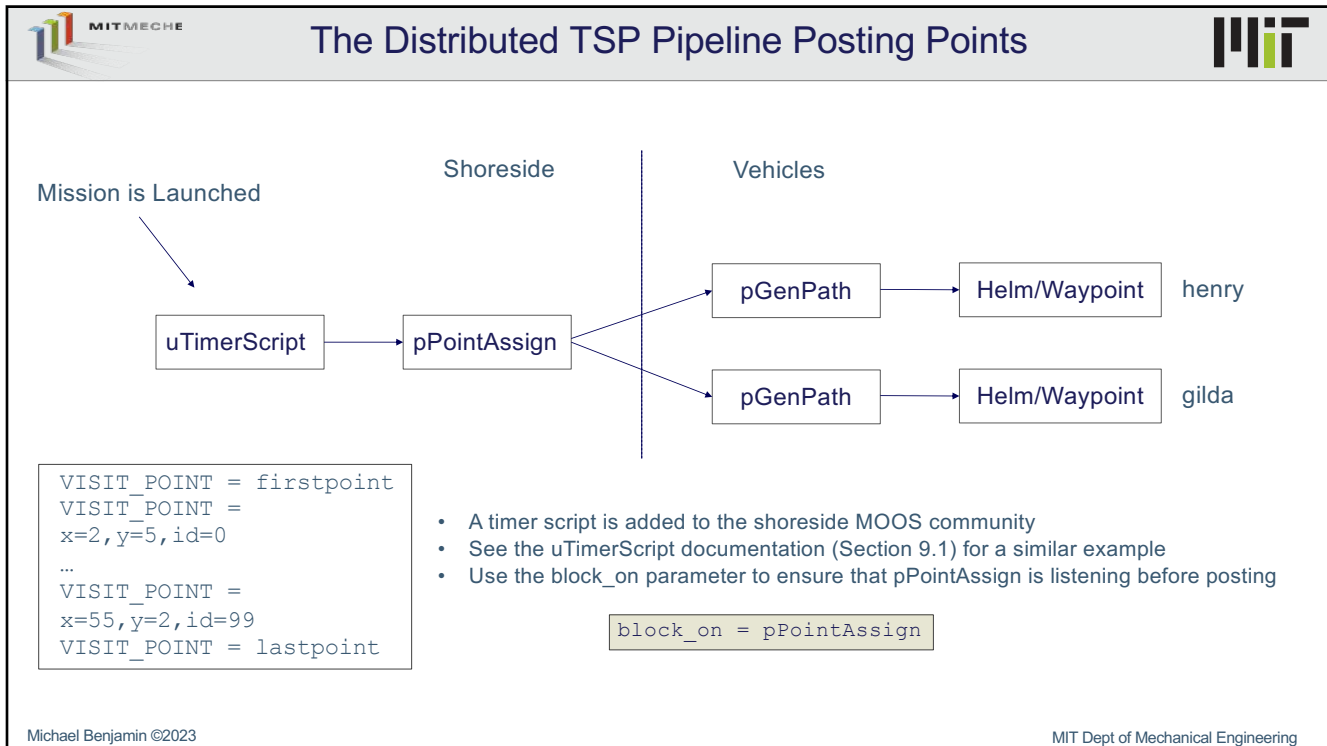
  paused     = false
  reset_time = all-posted
  reset_max  = 0
  block_on   = pPointAssign

  rand_var   = ...
  rand_var   = ...

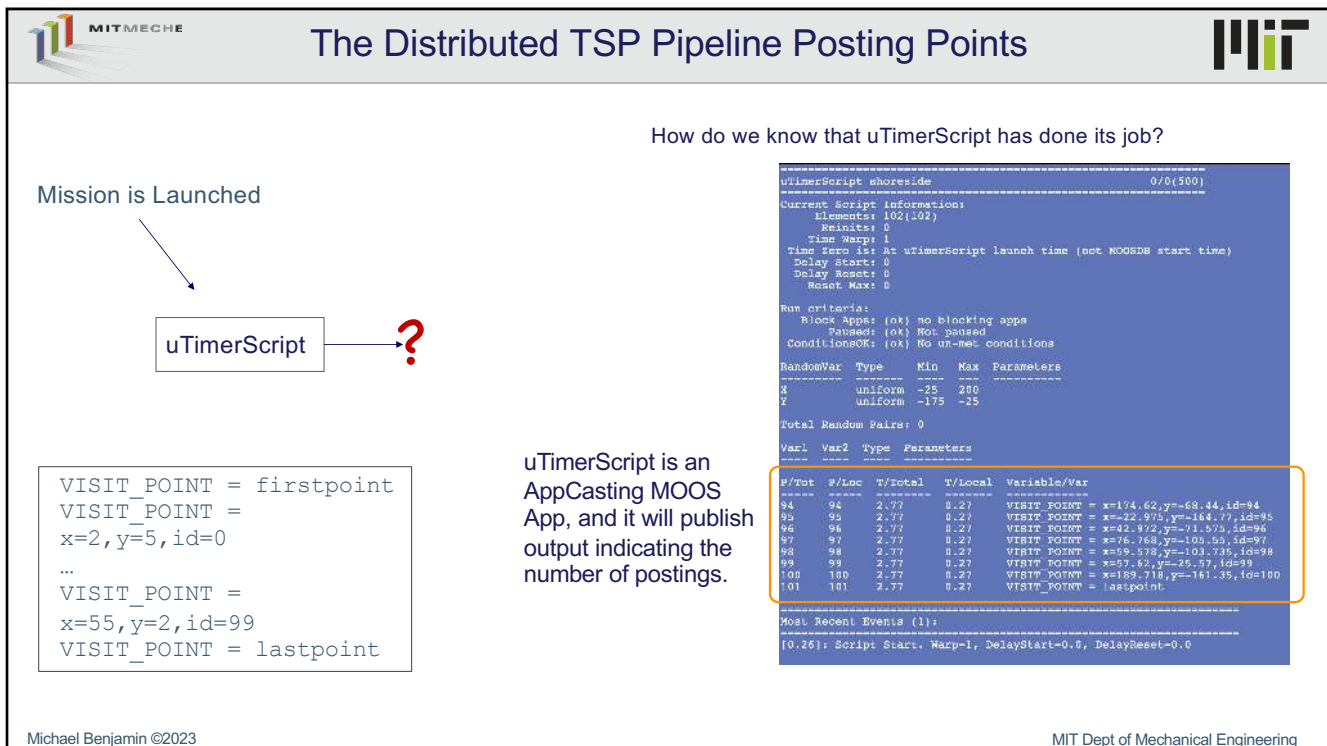
  event = ...
}
```

Michael Benjamin ©2023
MIT Dept of Mechanical Engineering


14




15



16



The Distributed TSP Pipeline Posting Points



Mission is Launched

uTimerScript

?

```

VISIT_POINT = firstpoint
VISIT_POINT =
x=2,y=5,id=0
...
VISIT_POINT =
x=55,y=2,id=99
VISIT_POINT = lastpoint
  
```


Assuming you are running pLogger, after the mission has been stopped, the aloggrep tool can be used to see all postings to a particular variable, or all postings made by a particular app:

```


$ aloggrep SHORESIDE.alog VISIT_POINT
=====
%% LOG FILE:      SHORESIDE.alog
%% FILE OPENED ON Wed Dec 31 19:00:00 1969
%% LOGSTART      6470836882.04
2.747    VISIT_POINT    uTimerScript    firstpoint
2.747    VISIT_POINT    uTimerScript    x=174.935,y=-171.595,id=1
2.747    VISIT_POINT    uTimerScript    x=-18.61,y=-119.68,id=2
2.747    VISIT_POINT    uTimerScript    x=137.428,y=-76.165,id=3
2.747    VISIT_POINT    uTimerScript    x=195.275,y=-128.785,id=4
2.747    VISIT_POINT    uTimerScript    x=179.435,y=-141.25,id=5
2.747    VISIT_POINT    uTimerScript    x=160.535,y=-94.015,id=6
2.747    VISIT_POINT    uTimerScript    x=184.137,y=-36.055,id=7
      o o o
2.753    VISIT_POINT    uTimerScript    x=174.62,y=-68.44,id=94
2.753    VISIT_POINT    uTimerScript    x=-22.975,y=-164.77,id=95
2.753    VISIT_POINT    uTimerScript    x=42.972,y=-71.575,id=96
2.753    VISIT_POINT    uTimerScript    x=76.768,y=-105.55,id=97
2.753    VISIT_POINT    uTimerScript    x=59.578,y=-103.735,id=98
2.753    VISIT_POINT    uTimerScript    x=57.62,y=-25.57,id=99
2.753    VISIT_POINT    uTimerScript    x=189.718,y=-161.35,id=100
2.753    VISIT_POINT    uTimerScript    lastpoint
  
```

Michael Benjamin ©2023 MIT Dept of Mechanical Engineering

17



The Distributed TSP Pipeline pPointAssign



Mission is Launched

Shoreside

uTimerScript

pPointAssign

How can we tell whether pPointAssign is working?

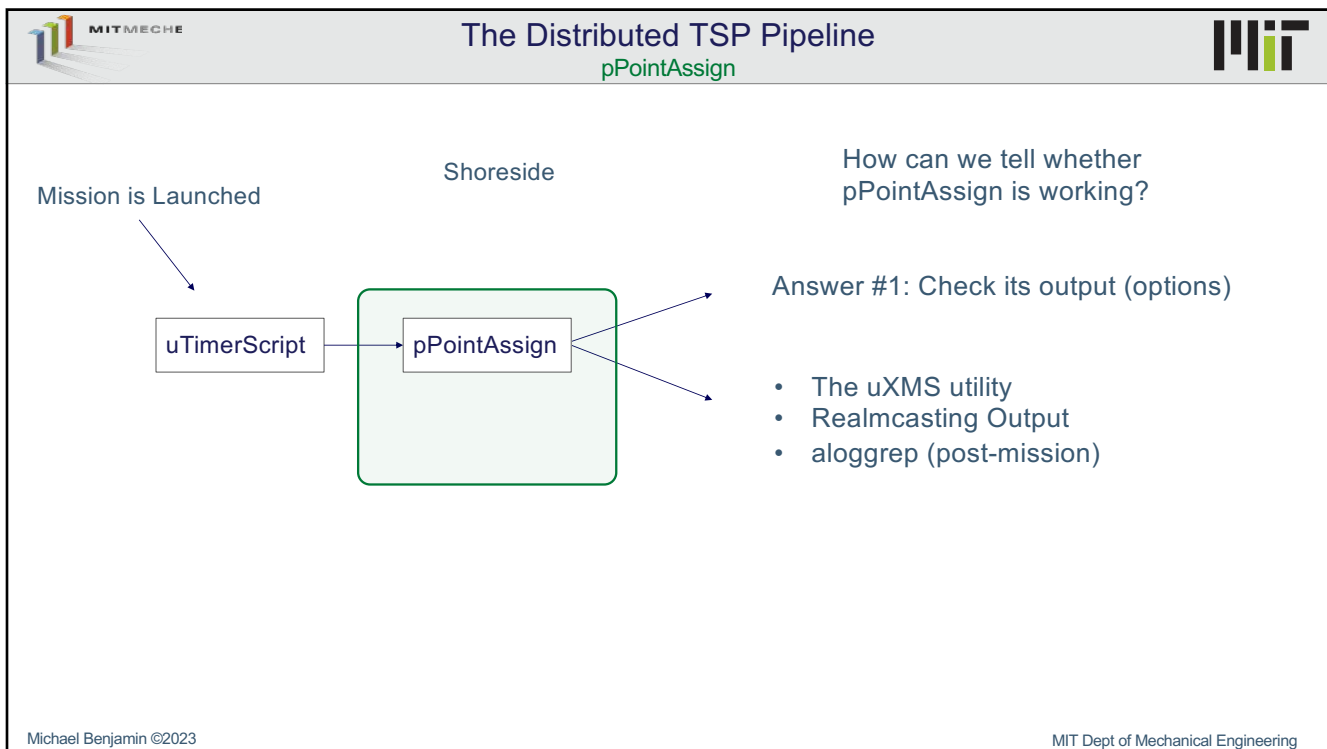
Answer #1: Check its output (options)

```

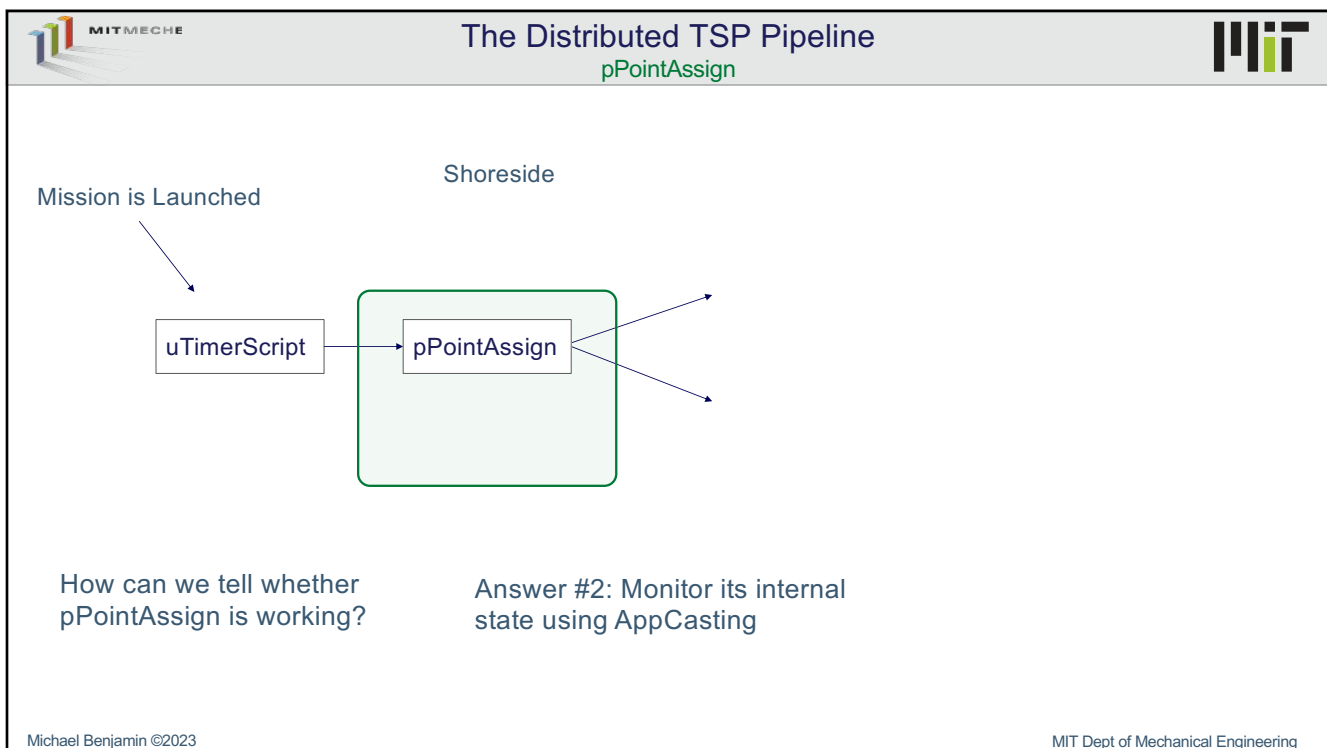
VISIT_POINT_HENRY =
firstpoint
VISIT_POINT_HENRY =
x=2,y=5,id=5
...
VISIT_POINT_GILDA =
firstpoint
VISIT_POINT_GILDA =
x=2,y=5,id=5
...
VISIT_POINT_GILDA =
x=55,y=2,id=96
VISIT_POINT_GILDA = lastpoint
  
```

Michael Benjamin ©2023 Engineering

18



19



20

Node	RC	App	RC
shoreside	58	MOOSDB shoreside	5
gilda	0	uMAC 8154	0
henry	0	VISIT_POINT	0
		VISIT_POINT GILDA	5
vstate	0	VISIT_POINT HENRY	48
		pLogger	0
		pMarineViewer	0
		pRealm	0
		uProcessWatch	0
		uTimerScript	0
		pHostInfo	0
		pPointAssign	0
		pShare	0
		uFldShoreBroker	0

MIT Dept of Mechanical Engineering




Gilda


Index	IC	App	IC
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	21	21	21
22	22	22	22
23	23	23	23
24	24	24	24
25	25	25	25
26	26	26	26
27	27	27	27
28	28	28	28
29	29	29	29
30	30	30	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	42
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	48
49	49	49	49
50	50	50	50
51	51	51	51
52	52	52	52
53	53	53	53
54	54	54	54
55	55	55	55
56	56	56	56
57	57	57	57
58	58	58	58
59	59	59	59
60	60	60	60
61	61	61	61
62	62	62	62
63	63	63	63
64	64	64	64
65	65	65	65
66	66	66	66
67	67	67	67
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	76	76	76
77	77	77	77
78	78	78	78
79	79	79	79
80	80	80	80
81	81	81	81
82	82	82	82
83	83	83	83
84	84	84	84
85	85	85	85
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	94
95	95	95	95
96	96	96	96
97	97	97	97
98	98	98	98
99	99	99	99
100	100	100	100

MIT Dept of Mechanical Engineering

11



The Distributed TSP Pipeline



pPointAssign

Mission is Launched

↓

uTimerScript

→

pPointAssign

Shoreside


Monitoring the output is great for **confirming** that things work, but what if things are not working and there is no output?

How can we tell whether pPointAssign is working?


Answer #2: Monitor its internal state using AppCasting

Michael Benjamin ©2023 MIT Dept of Mechanical Engineering

23



The Distributed TSP Pipeline



pPointAssign

Mission is Launched

↓

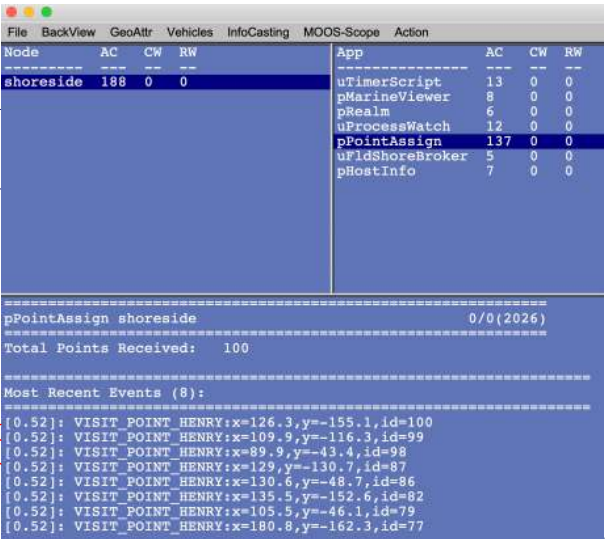
uTimerScript

→

pPointAssign

Shoreside

Example AppCasting output for pPointAssign

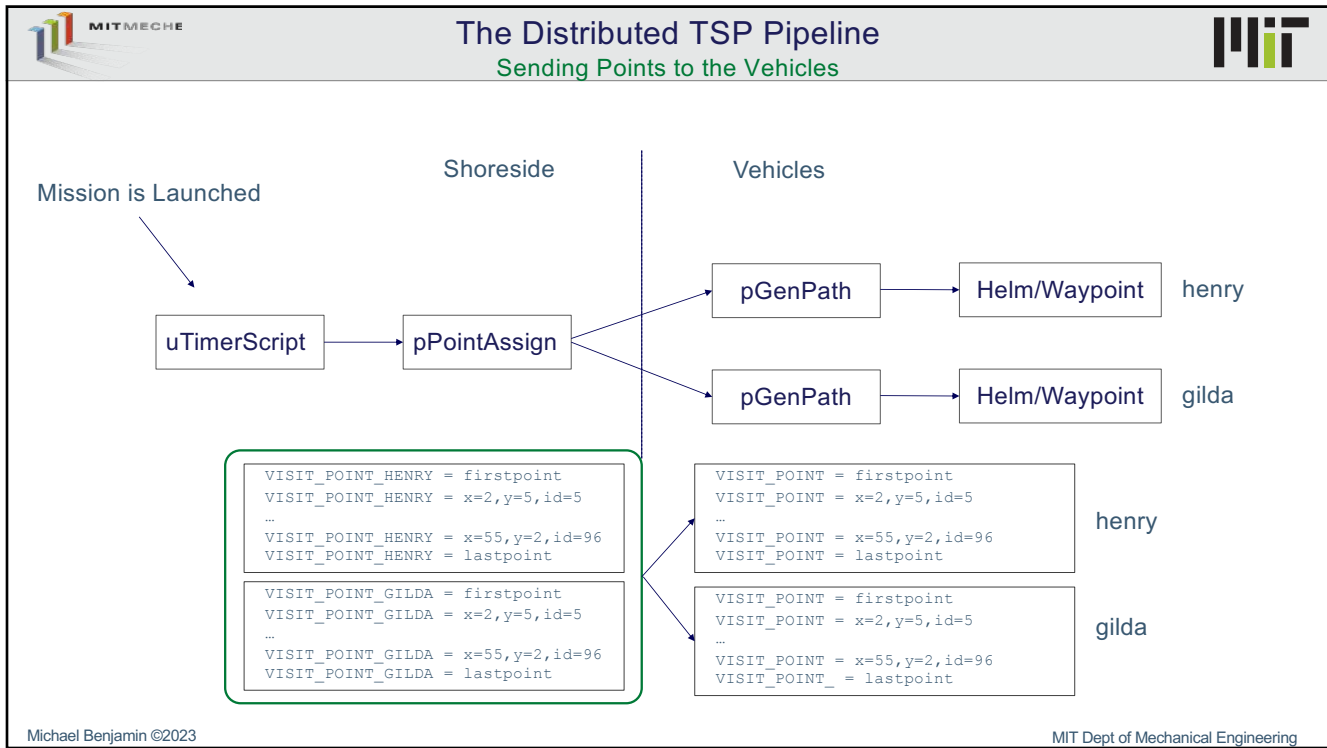


- Confirm points received
- Confirm postings
- You can add whatever helps

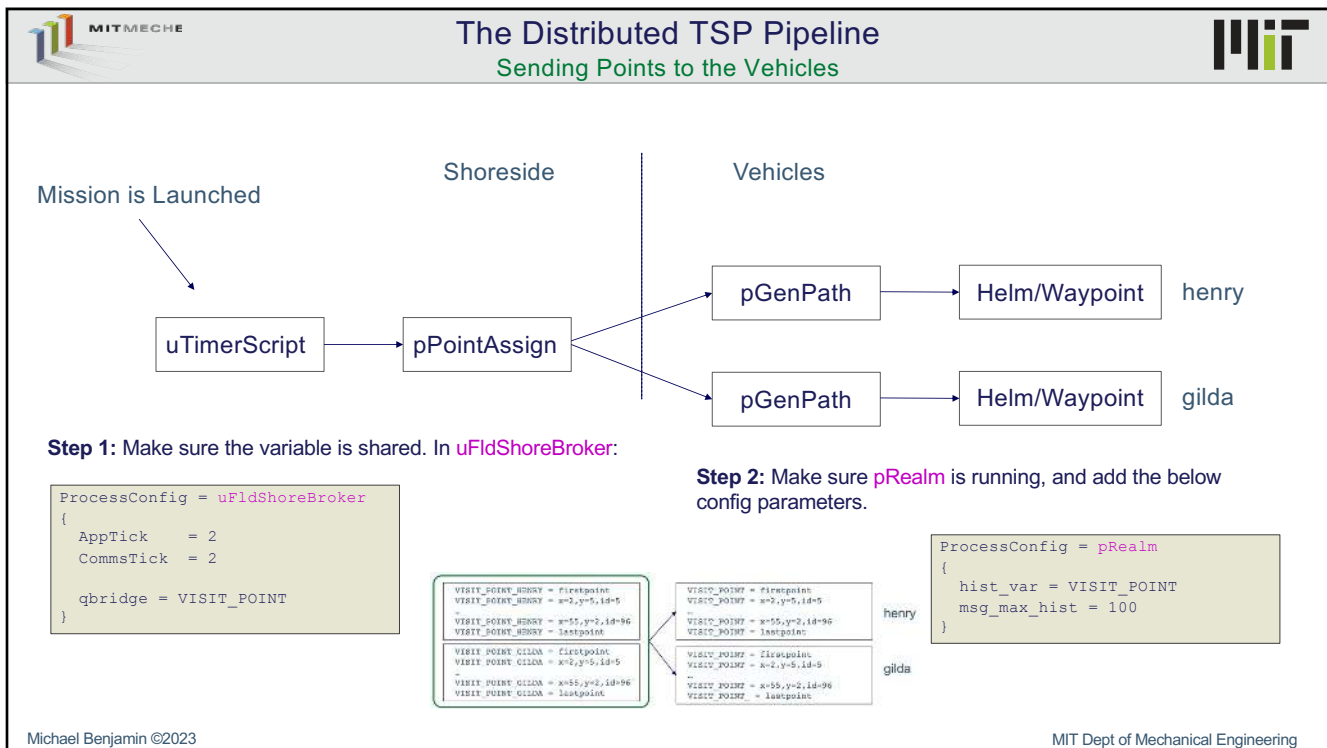
Red arrows point from the list to the 'Total Points Received' and 'Most Recent Events' sections of the AppCasting output.

Michael Benjamin ©2023 MIT Dept of Mechanical Engineering

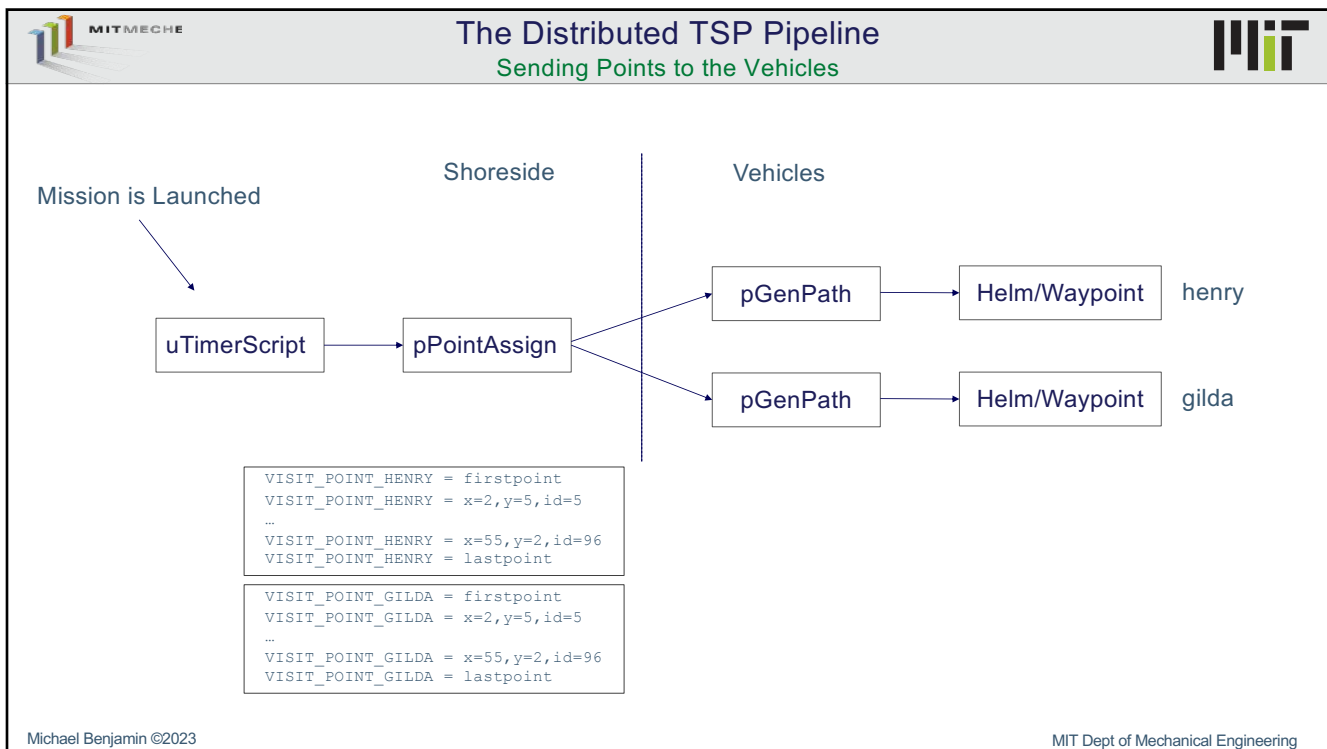
24



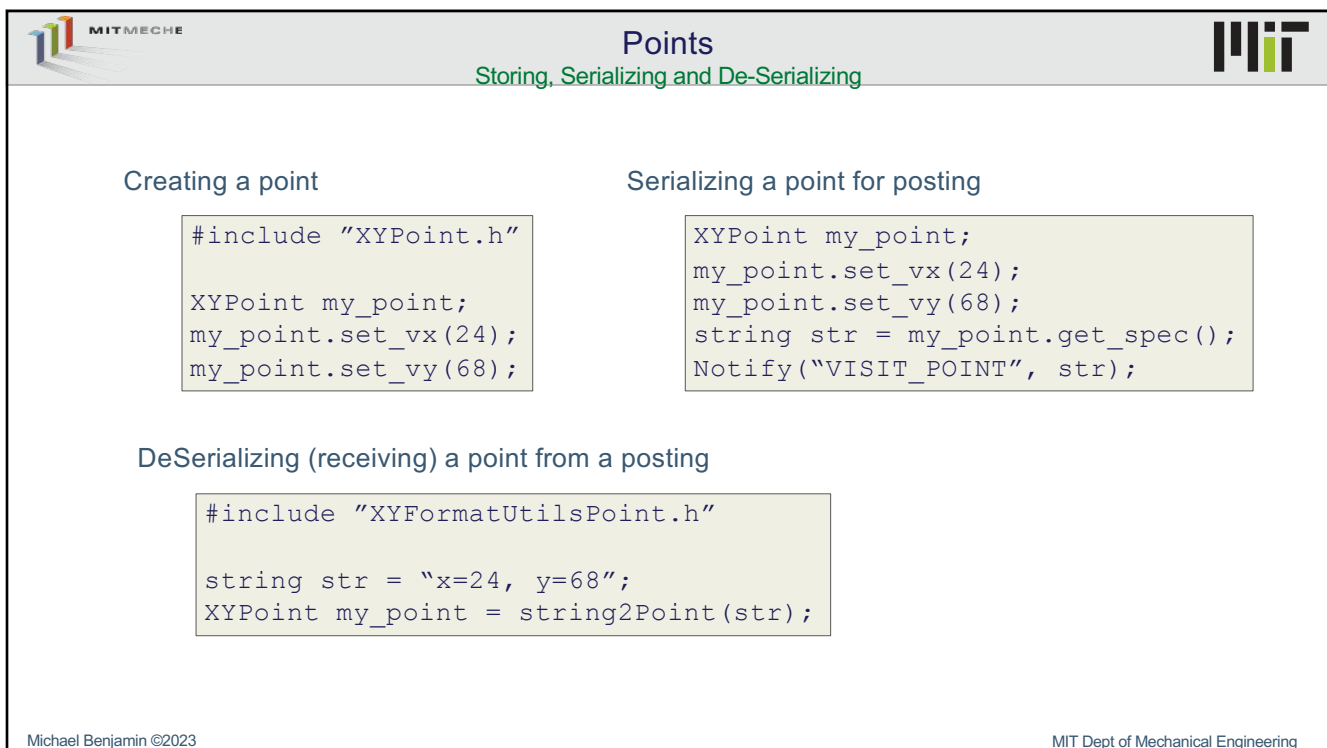
25




26



27



28



END

Michael Benjamin ©2023

MIT Dept of Mechanical Engineering