

# MSE Presentation 1

## MultiAgent Control of Traffic Signals (MACTS)

Bryan Nehl

MSE Candidate



# Agenda

- Project Vision
- System Context
- Project Requirements and Plan
- Cost Estimation
- Project Software Quality Assurance Plan
- Candidate Architecture Overview
- Demonstration Agent Communication
- Demonstration Interaction with Simulator (SUMO)
- Phase II Deliverables
- Project Log, Risks, Spikes and Concerns
- Questions/Comments
- References



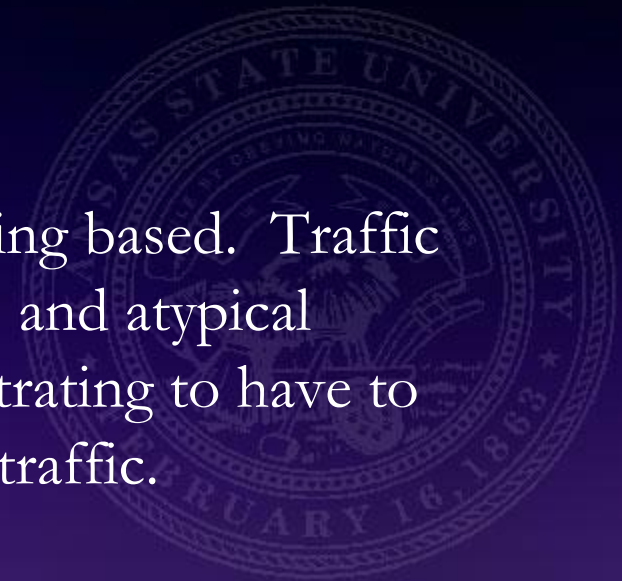
# Project Vision

- **Goal**

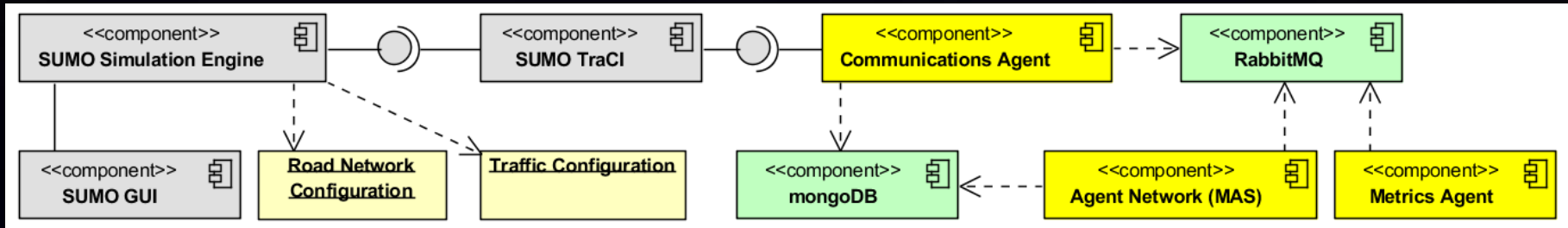
The goal of this project is to create a multiagent system that is capable of traffic light signal control which results in an improved travel experience.

- **Motivation**

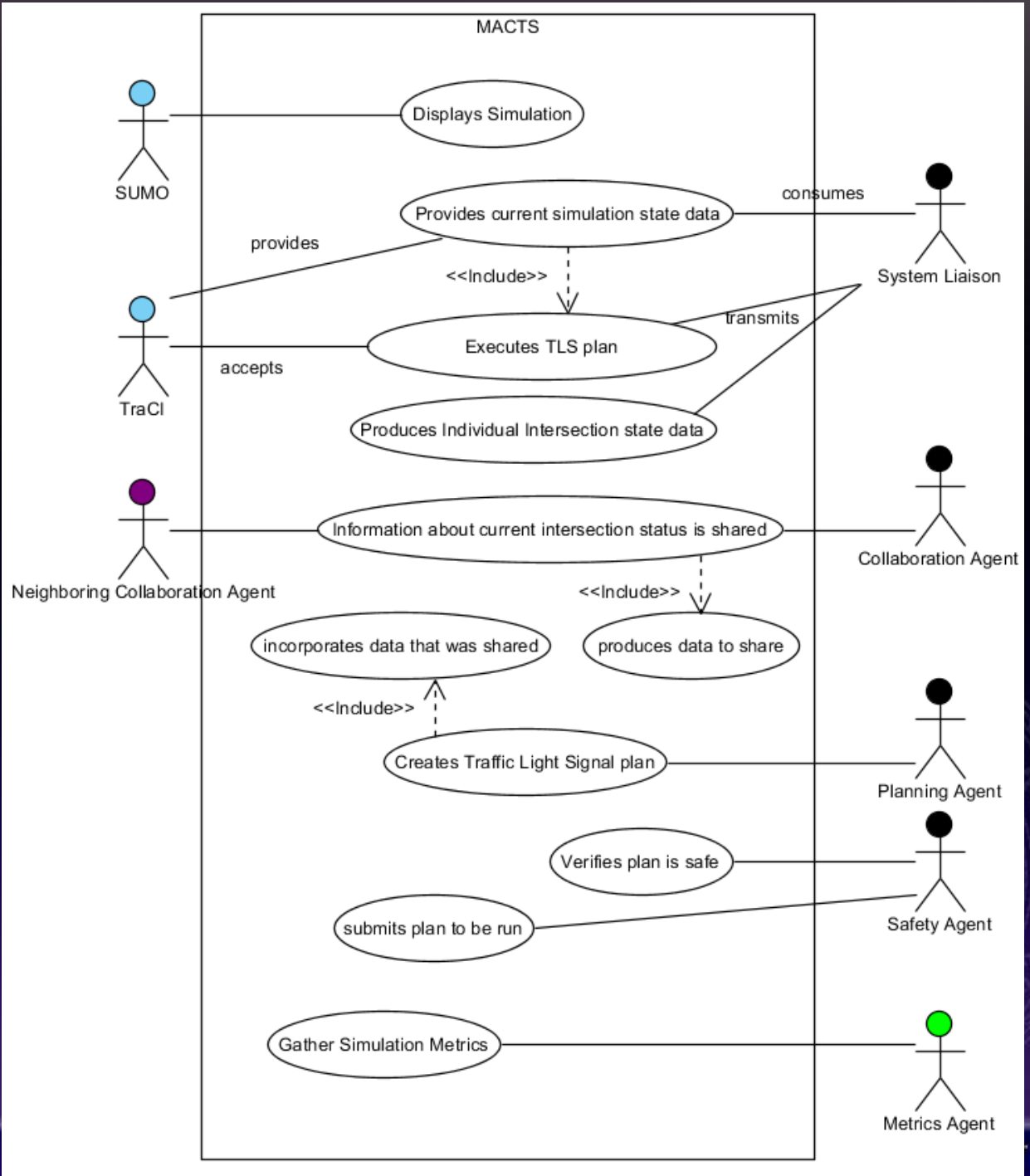
Most traffic light systems today are strictly timing based. Traffic flow studies are required to create timing plans and atypical conditions cause problems. It is also very frustrating to have to stop for a red light when there is no opposing traffic.



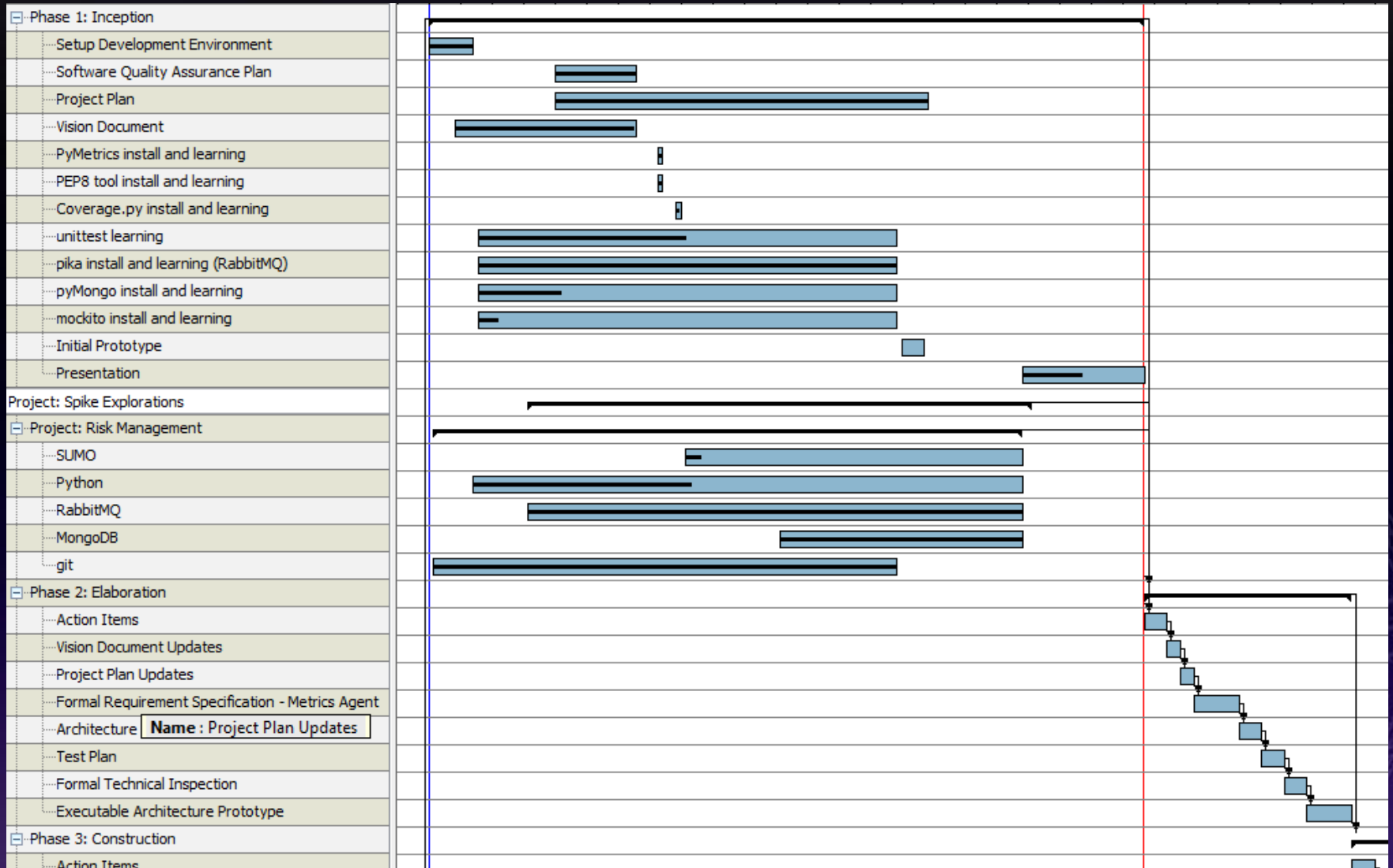
# System Context



# Project Requirements



# Project Plan



# Cost Estimation

This model is explained in Section 4 of the COCOMO II Design Model Definition document [8].

(Equation 1)

$$TDEV = [C \times (PM_{NS})^F] \times \frac{SCED\%}{100}$$

$$F = (D + 0.2 \times [E - B])$$

Symbol	Description
B	The scaling base-exponent for the effort equation, currently set to 0.91
C	Coefficient that can be calibrated currently set to 3.67
D	Scaling base-exponent that can be calibrated currently set to 0.28
E	The scaling exponent for the effort equation
$PM_{NS}$	Person-Months estimated without the SCED cost driver (Nominal Schedule)
SCED	Required Schedule Compression
TDEV	Time to Develop in calendar months



### 2.1.2 Estimate Variables

Cost Driver	Value (Text)	Factor	Description
LANG	Hybrid 3 <sup>rd</sup> /4 <sup>th</sup> Python	50	Used for converting from UFP to SLOC
PREC	Nominal	3.72	<u>Precedentedness</u>
FLEX	High	2.03	Development Flexibility
RELY	Very High	5	Required Software Reliability
DATA	Low	2	Data Size
CPLX	Nominal	4	Product Complexity
RUSE	Low	0.95	Required reusability
DOCU	Nominal	3	Documentation to match life cycle needs
RESL	High	1.41	Architecture and risk resolution
TEAM	High	2.19	Team Cohesion
ACAP	High	4	Analyst Capability
PCAP	High	4	Programmer Capability
PCON	Very High	5	Personnel Capability
APEX	Nominal	3	Application Experience
PLEX	Nominal	3	Platform Experience
LTEX	Low	2	Language and Tool Experience
PMAT	High	3.12	Process Maturity
TIME	Very High	5	Execution Time
STOR	Nominal	3	Main Storage Constraint
PVOL	Low	2	Platform Volatility
TOOL	Nominal	3	Use of Software Tools
SITE	Extra High	6	Multisite Development
SCED	Nominal	1.00	Schedule



# Cost Estimation: Size Estimates

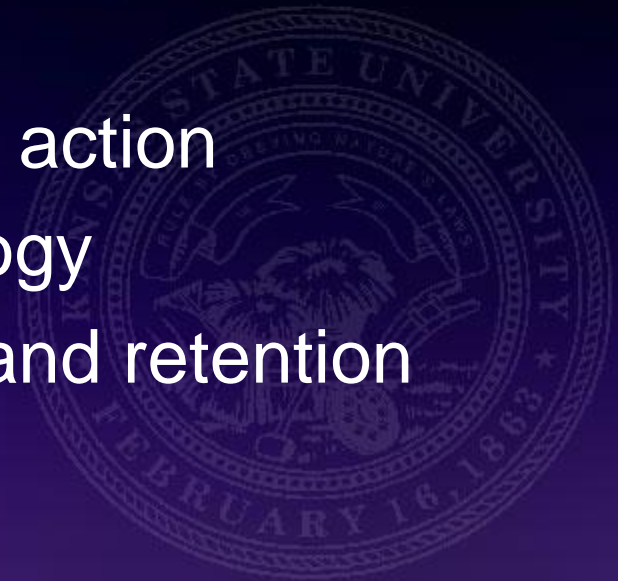
Use Case	ILF	EIF	EI	EO	EQ	Total	
1		2			1		
2					2	2	
3		1			1		
4		1			1		
5			1	1	1	1	
6		1					
7		1	1		1	1	
8		1			1		
9				1	1		
10		1			1		
11		1	1		1	1	
<b>Sub Total</b>		9	3	2	11	5	
<b>All weights</b>		7	5	3	4	3	
<b>Low weighted</b>		63	15	6	44	15	128

# Early Design Calculations

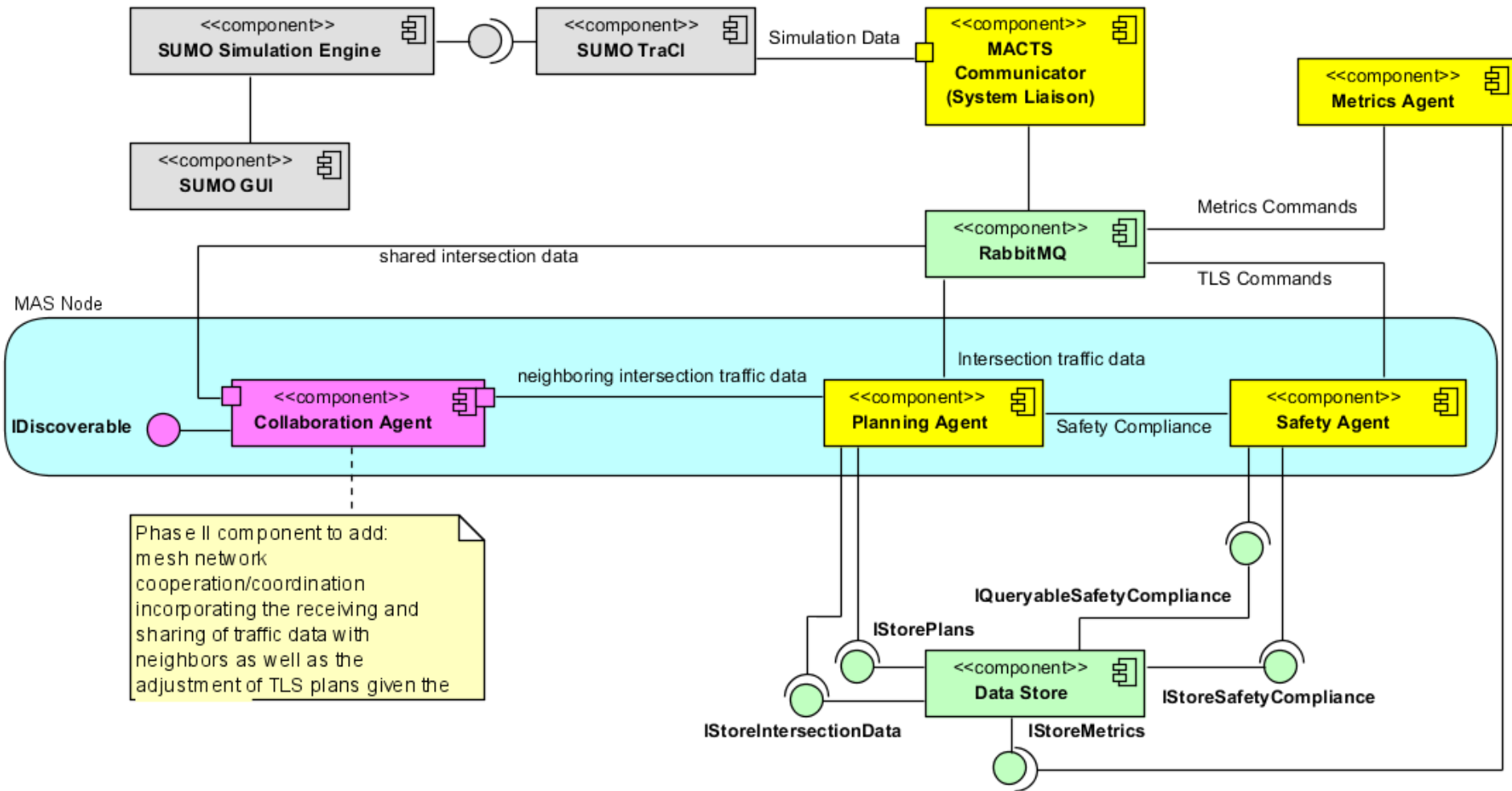
RCPX	1.33	RELY + DATA + CPLX + DOCU Product Reliability and Complexity
RUSE	0.95	RUSE Developed for Reusability
PDIF	1.29	TIME + STOR + PVOL Platform Difficulty
PREX	1.12	APEX + PLEX + LTEX Personnel Experience
PERS	0.63	ACAP + PCAP + PCON Personnel Capability
FCIL	0.73	TOOL + SITE Facilities
SCED	1.00	SCED Required Development Schedule
<u>E<sub>Arch</sub></u>	0.8395	Product of the above defined composite cost drivers.
Size (KSLOC)	6.4	
Size (UFP)	128	Unadjusted Function Points
UFP->Lines of Code	50	Conversion factor from UFP to SLOC
PREC	0.01	<u>Precedentedness</u>
FLEX	0.03	Flexibility
RESL	0.03	Risk Resolution
TEAM	0.04	Team cohesiveness and communication
PMAT	0.03	Process Maturity
Process Exponent	1.15	(A sum of the PREC, FLEX, RESL, TEAM and PMAT parameters)
Effort	17.39	staff-months
TDEV	2.82	time to develop C=3.0, D=0.33, B=1.01
TDEV Early Design	3.47	time to develop C=3.67, D=0.28, B=0.91
TDEV 1997 Calibration	2.50	time to develop C=2.66, D=0.33, B=1.01

# Project Software Quality Assurance Plan (SQAP)

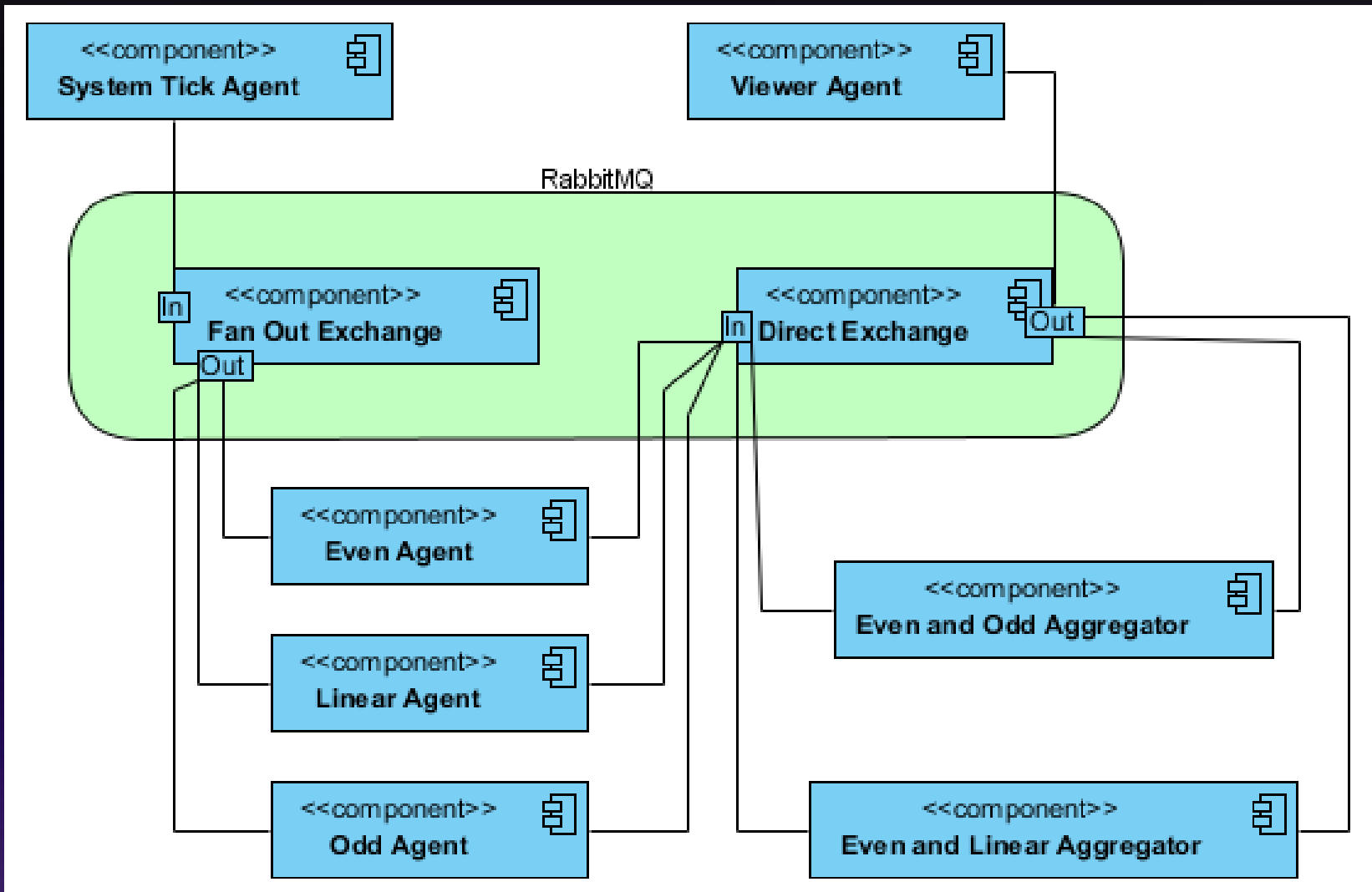
- Management Organization
- Documentation
- Standards, Practices, Conventions and Metrics
- Test plan
- Problem reporting and corrective action
- Tools, techniques and methodology
- Record collection, maintenance and retention



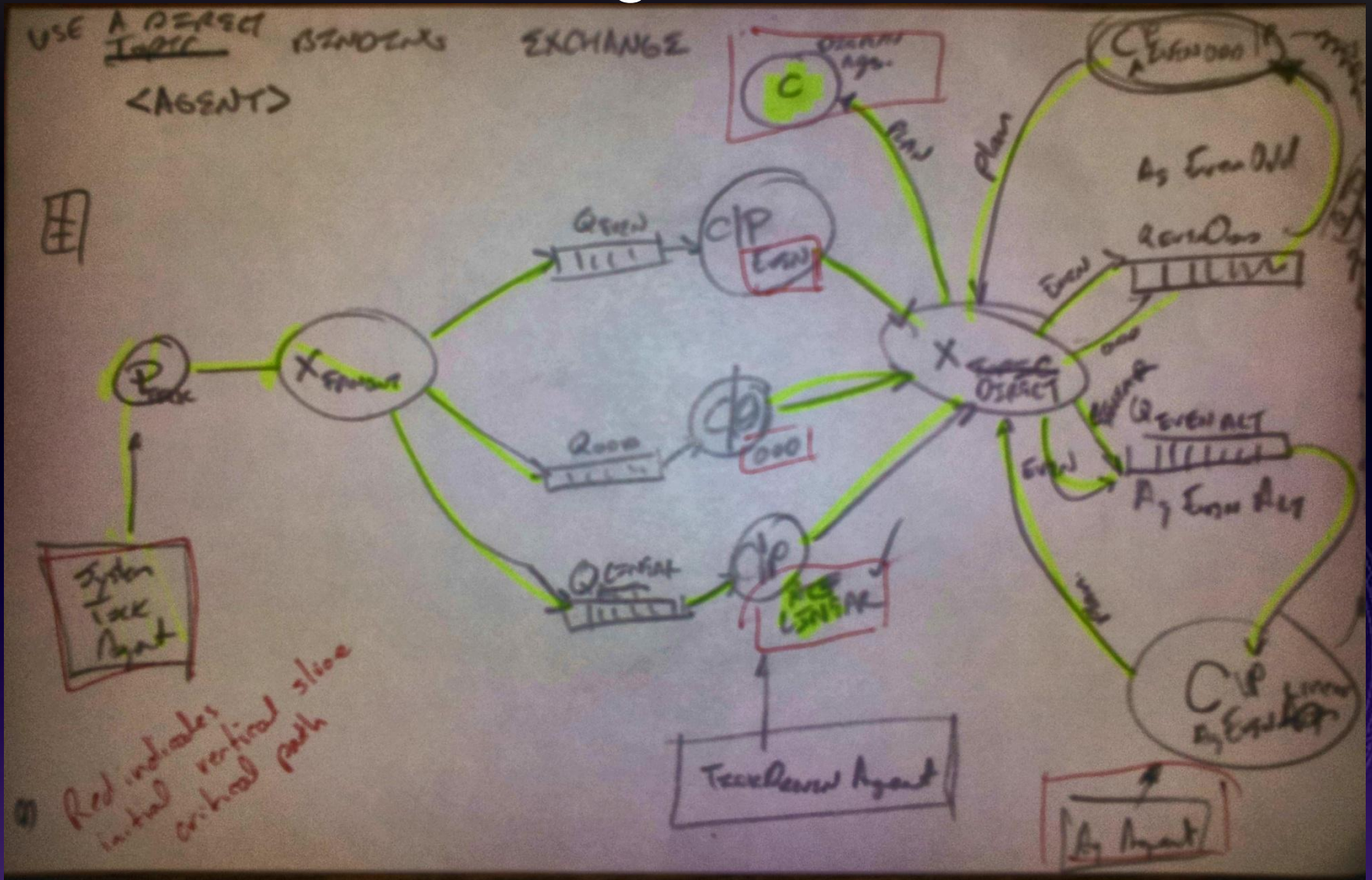
# Candidate Architecture Overview



# Demonstration: Agent Communication

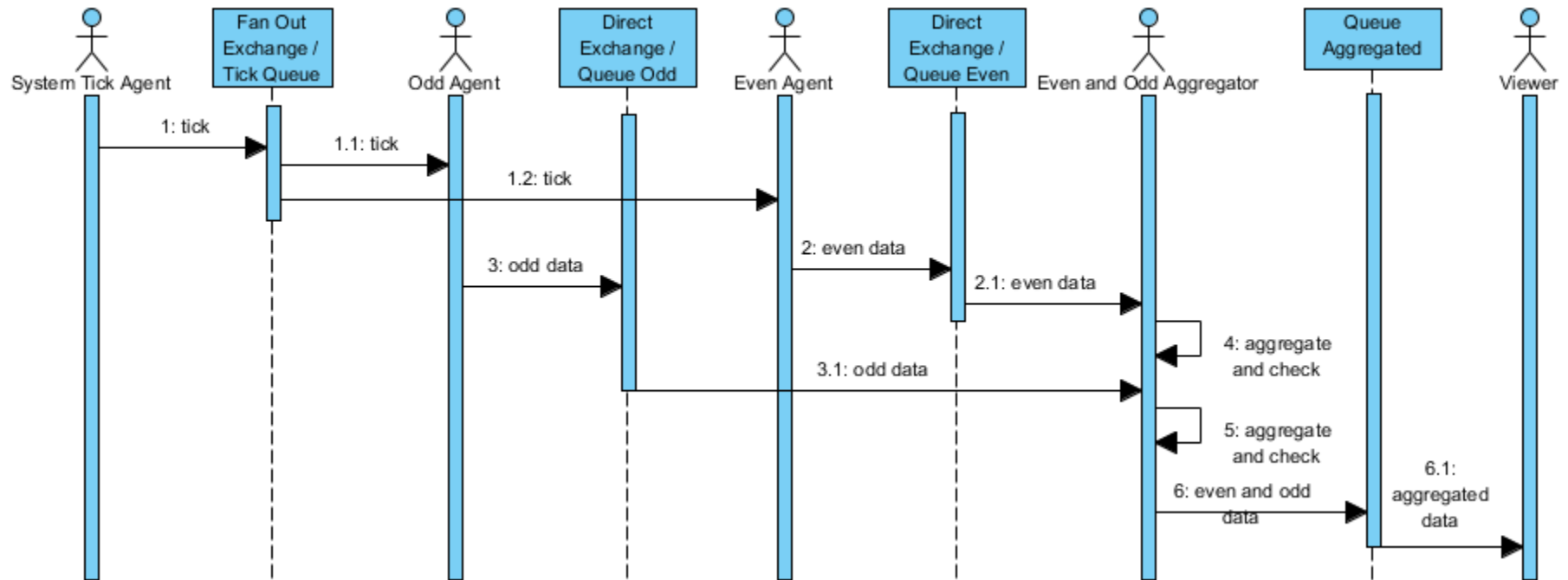


# Demonstration Agent Communication

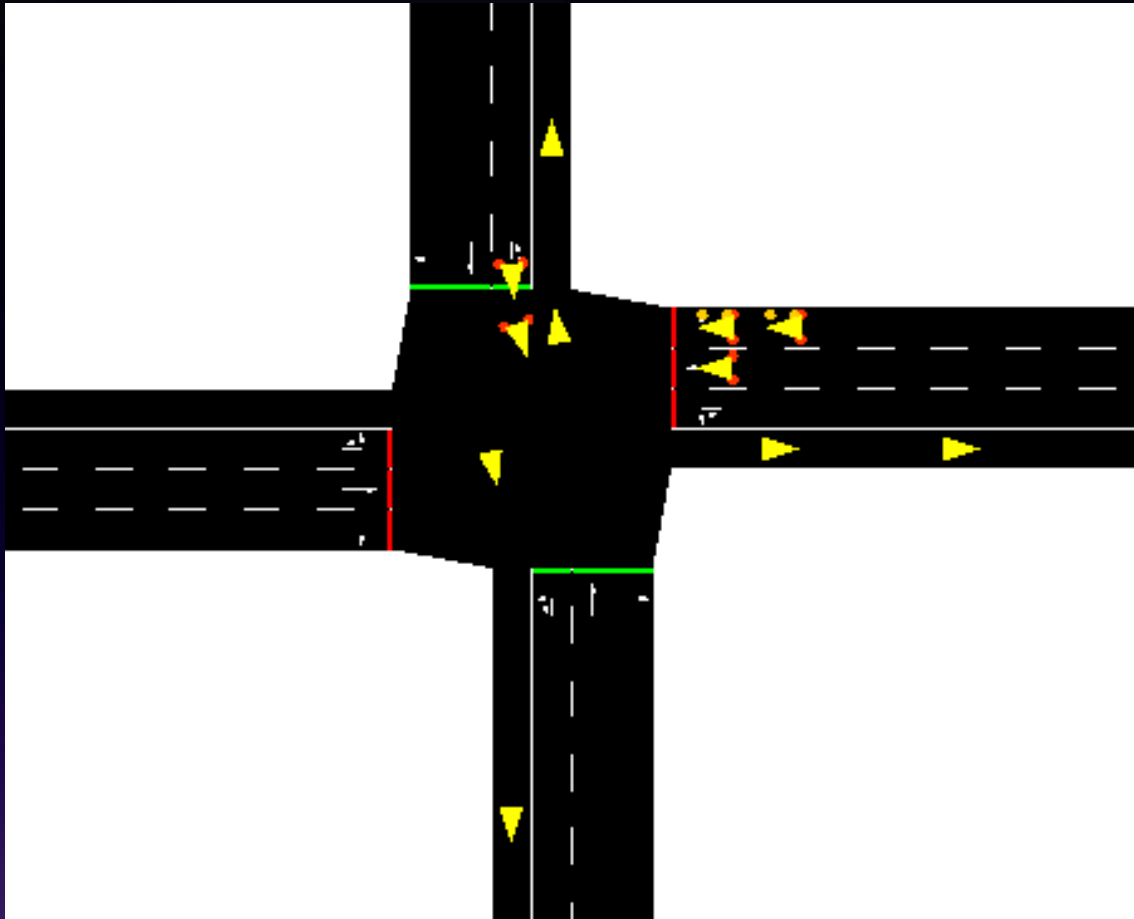


# Demonstration Agent Communication

sd Aggregate



# Demonstration Interaction with SUMO





# Phase II Deliverables

- Action Items from Phase 1
- Vision Document 2.0
- Project Plan 2.0
- Formal Requirements Specification
- Architecture Design 1.0
- Test Plan
- Formal Technical Inspection Checklist
- Executable Architecture Prototype
- Risk Log Update
- Presentation 2



# Project Log

- Maintained as a Google Document
- Exported as PDF for website



# Project Risks and Spikes

## Risks

- Python
- RabbitMQ
- MongoDB
- SUMO
- git
- Scope
- Time

## Spikes

- Python
- RabbitMQ
  - Q Fanout
  - Aggregation
- MongoDB
  - training
- git
  - reading
  - seminars
  - hands on use

## Spikes

- SUMO
  - Network Load
  - Network Double T
  - Read from TRACI
  - Send to TRACI
  - Network Metrics
  - Read Sensors
  - Add Sensors



# Questions and Comments



# References

- SUMO, “Simulation for Urban MObility,” Sep. 2011; [http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Main\\_Page](http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Main_Page).
- T. Masterton and D. Topiwala, “Multi-Agent Traffic Light Optimisation and Coordination,” white paper, Thales Group, Reference VCS081002, Issue 2, 2008.
- W. Royce, Software Project Management: A Unified Framework, Addison-Wesley, 1998, p. 34, pp. 265-281.
- B. Boehm et al., “Cost Models for Future Software Processes: COCOMO 2.0,” Annals of Software Eng., Vol. 1, 1995, pp. 57-94.
- K-State Master of Software Engineering web site, “MSE Portfolio Requirements,”
- November 28, 2011; <http://mse.cis.ksu.edu/portfolio.html>.
- Center for Systems and Software Engineering web site, “COCOMO II,” December 4, 2011: [http://sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html).
- The Code Project web site, “Software Project Cost Estimates Using COCOMO II Model,” December 4, 2011: <http://www.codeproject.com/KB/architecture/cocomo2.aspx>.
- Naval Postgraduate School web site, “COCOMO II - Constructive Cost Model,” December 4, 2011: <http://diana.nps.edu/~madachy/tools/COCOMOII.php>.
- Center for Software Engineering, USC, COCOMO II: Model Definition Manual Version 2.1, 2000.
- The Code Project web site, “Calculating Function Points,” December 4, 2011: [http://www.codeproject.com/KB/architecture/Calculate\\_Function\\_Point.aspx](http://www.codeproject.com/KB/architecture/Calculate_Function_Point.aspx).
- USC Center for Software Engineering website, “COCOMO II Affiliates,” December 4, 2011: [http://csse.usc.edu/csse/affiliate/private/COCOMOII\\_Driver+Calc\\_Ss/SpreadSheet-COCOMOII.html](http://csse.usc.edu/csse/affiliate/private/COCOMOII_Driver+Calc_Ss/SpreadSheet-COCOMOII.html).

# References 2

- IEEE Std. 730-1998, IEEE Standard for Software Quality Assurance Plans, IEEE 1998.
- IEEE Std. 730.1-1995 IEEE Guide for Software Quality Assurance Planning, IEEE, 1995.
- Python Software Foundation, “PEP 8 -- Style Guide for Python Code”, Python, 24 Sep. 2011; <http://www.python.org/dev/peps/pep-0008/>.
- Python Software Foundation, “PEP 257 – Docstring Conventions”, Python, 24 Sep. 2011; <http://www.python.org/dev/peps/pep-0257/>.
- K. Hill, “GMoDS-based Runtime Agent Role Interpreter SQA Plan 1.0”, People, 15 Sep. 2011; [http://people.cis.ksu.edu/~kylhill/phase\\_1/sqa\\_plan.pdf](http://people.cis.ksu.edu/~kylhill/phase_1/sqa_plan.pdf).
- B. Nehl, “Multiagent Control of Traffic Signals Project Plan 1.0”, People, 26 Sep. 2011; <http://people.cis.ksu.edu/~bnehl/repos/macts.git/>.
- W. Royce, Software Project Management; Addison-Wesley, 1998, pp. 290-291.

