ANODE

Design and Implementation of an Extensible Platform for Large Scale, Programmatic Transmission Outage Planning Analysis

2018 Grid of the Future Symposium
October 29-30
Reston, VA

Kevin D. Jones, Ph.D. & Matthew A. Parker
Dominion Energy
USA

Mojtaba Jalalpour
University of Tennessee, Knoxville
USA

Overview: Outage Planning @ Dominion Energy

Status Quo

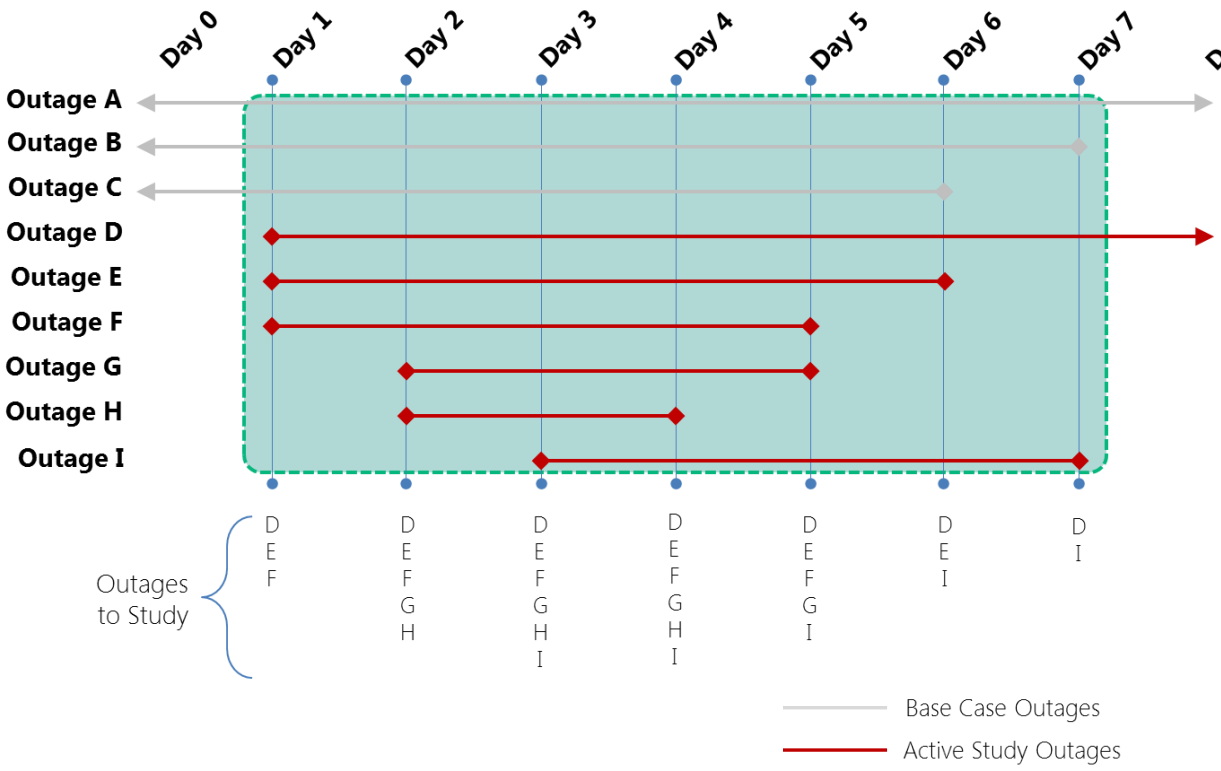
- Manual power-flow and N-1 with human derived corrective actions in the EMS
  - *Day Ahead*
  - *Ten Day Outlook*
  - *One Month Outlook*
  - *Two Month Outlook*
  - *Ad-hoc Long Term Analysis*

Challenges

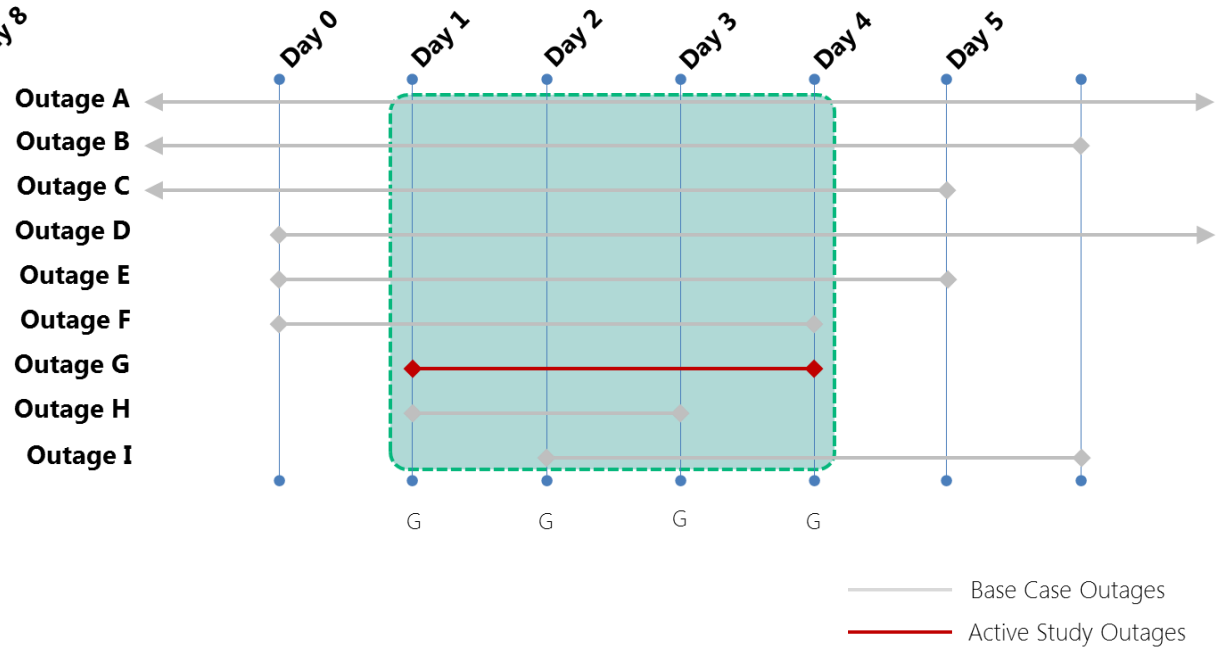
- Industry Challenges
  - *Retiring Generation*
  - *Aging Infrastructure*
  - *New Technologies*
  - *Renewables*
- Operations Planning Challenges
  - *How do you study an unfamiliar system?*
  - *Limitations of existing tools prevent scaling*

Problem Formulation: *Outage Planning Base Classes*

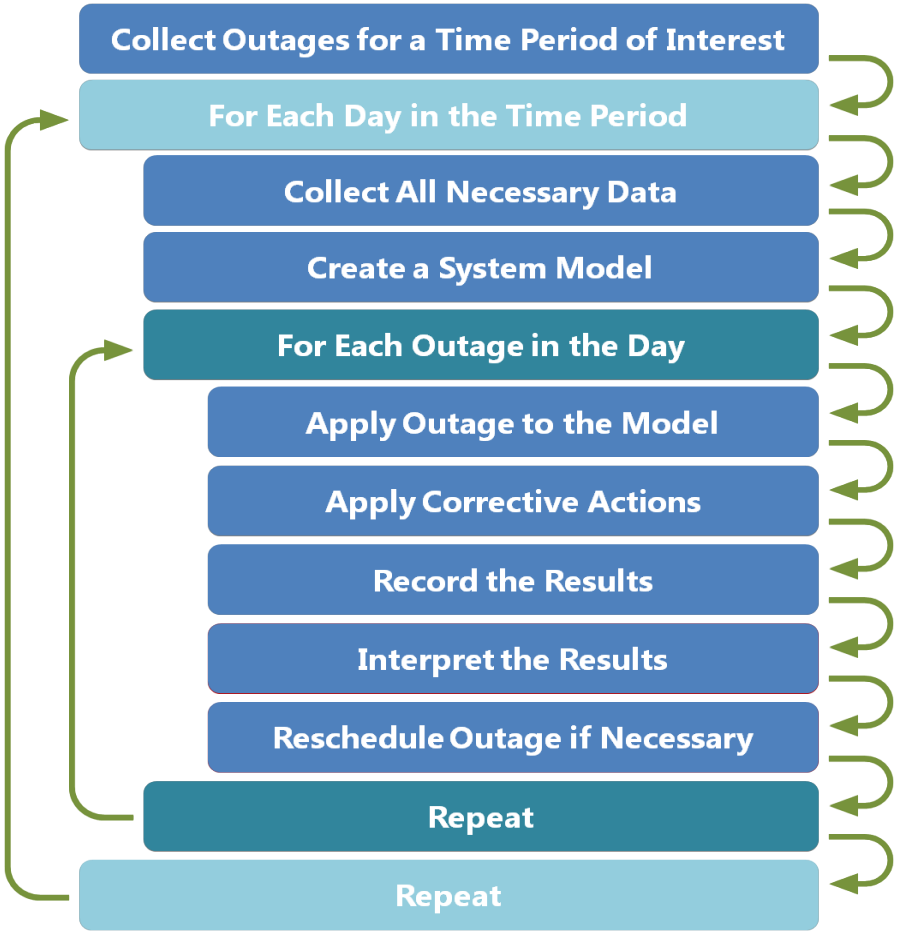
Time Period Study



Incremental Outage Study



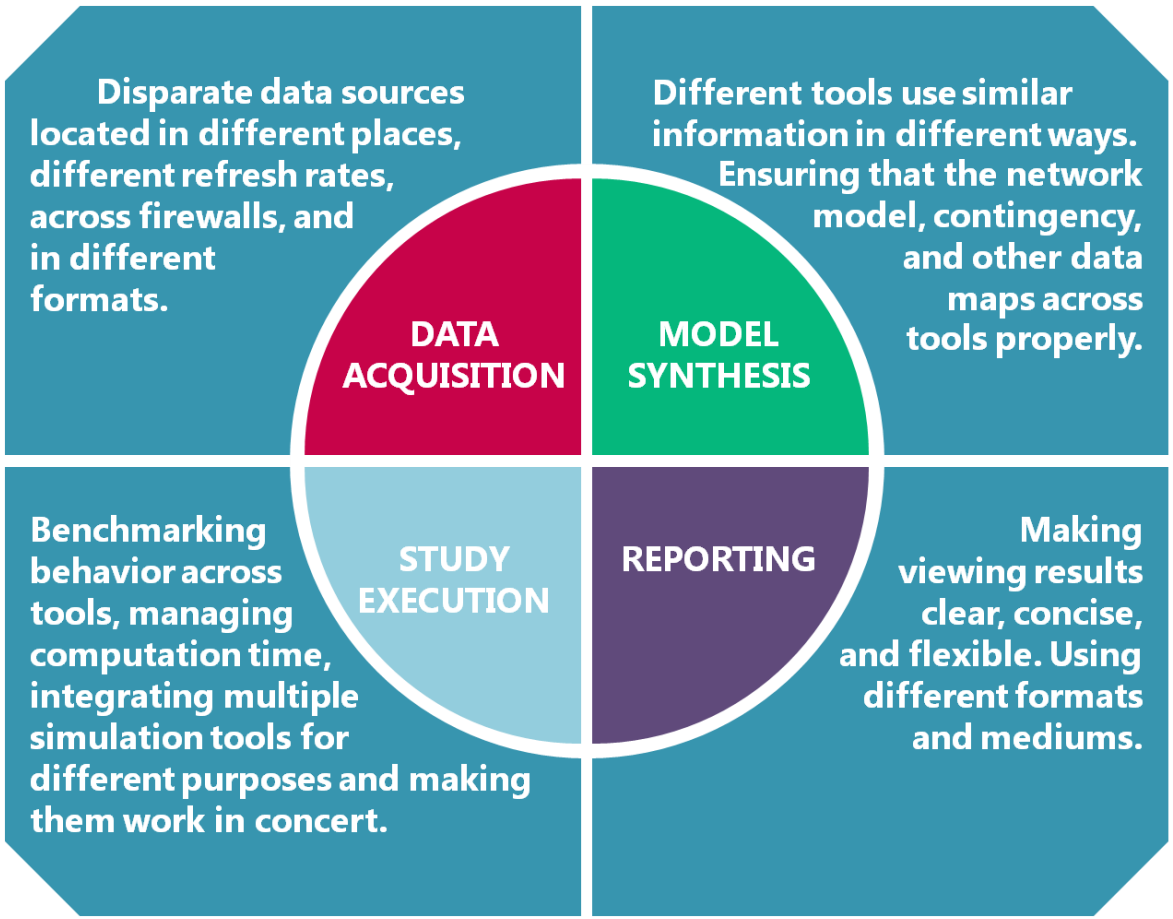
Problem Formulation: *Potential for Automation*



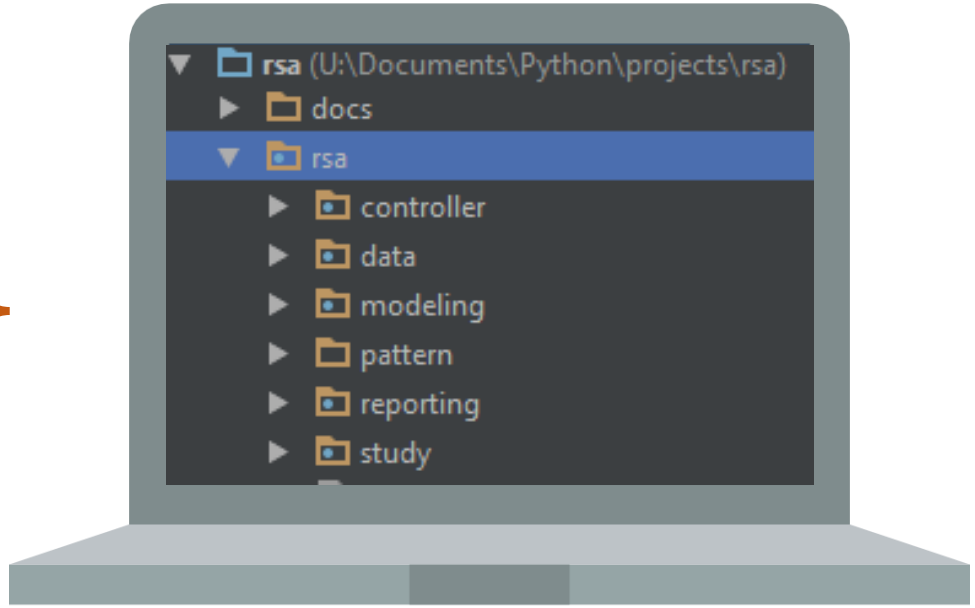

```
for each day in time_period:
    for each outage in day:
        outage.run_analysis()
        if outage.OK:
            outage.include_in_base_case = true
            outage.report_corrective_actions()
        else:
            outage.report_violations()
```

ANODE

Problem Formulation: *Four Domains*



These core concepts are reflected in software architecture



Design

Technology Selection, Data Sources, Model Synthesis,

ANODE

System Design: *Technology Selection*

Approaches

- Existing Vendor Solution
  - *Only siloed technologies exist*
- Custom Vendor Solution
  - *Requirements were vague*
- EMS Extension
  - *Incompatible technologies*
- Desktop Application
  - *This is a systems integration problem*
- In-House Custom Solution

Components

- Siemens PSSE – fine modeling
- PowerGEM TARA – bulk compute
- GE-Alstom EMS – data source
- hdbexport, csv – export mechanism
- Windows Task Scheduler – automation
- HP Superdome – compute
- Network Mounted HDD – storage
- Python – integration, orchestration, APIs

ANODE

System Design: *Data Sources*

System Design: *Model Synthesis*

1. **EMS Export** – Every 10 minutes SE Snapshot
2. **Export Package** - *.sav, *.raw, *.con, *.mon, *.sub
3. **Seed Case** – Optimally selected from Export Package Archive
4. **All In Case** – Normal Topology, Market Results for Units, No PV, Cleanup
5. **Alternate Limit Cases** – A case is created for each limit set in the study
6. **Base Cases** – Alternate Limit Cases handed off to TARA AMB for load scaling, SCRD, corrective actions
7. **Final Approved** – Daily cases with all outages present in topology, corrective actions present.

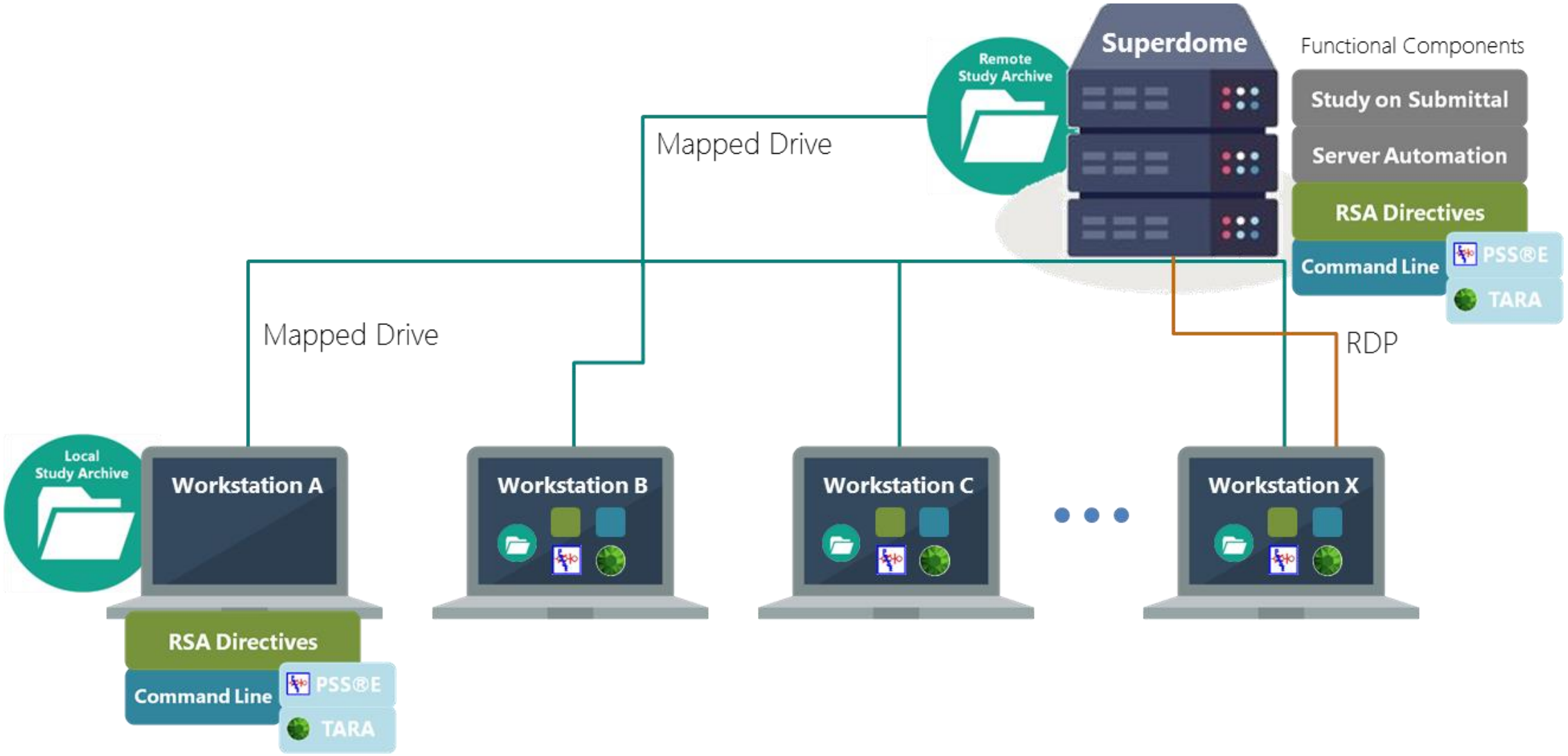
ANODE

Architecture

Hardware, Software, API, Automation

ANODE

Architecture: *Hardware*



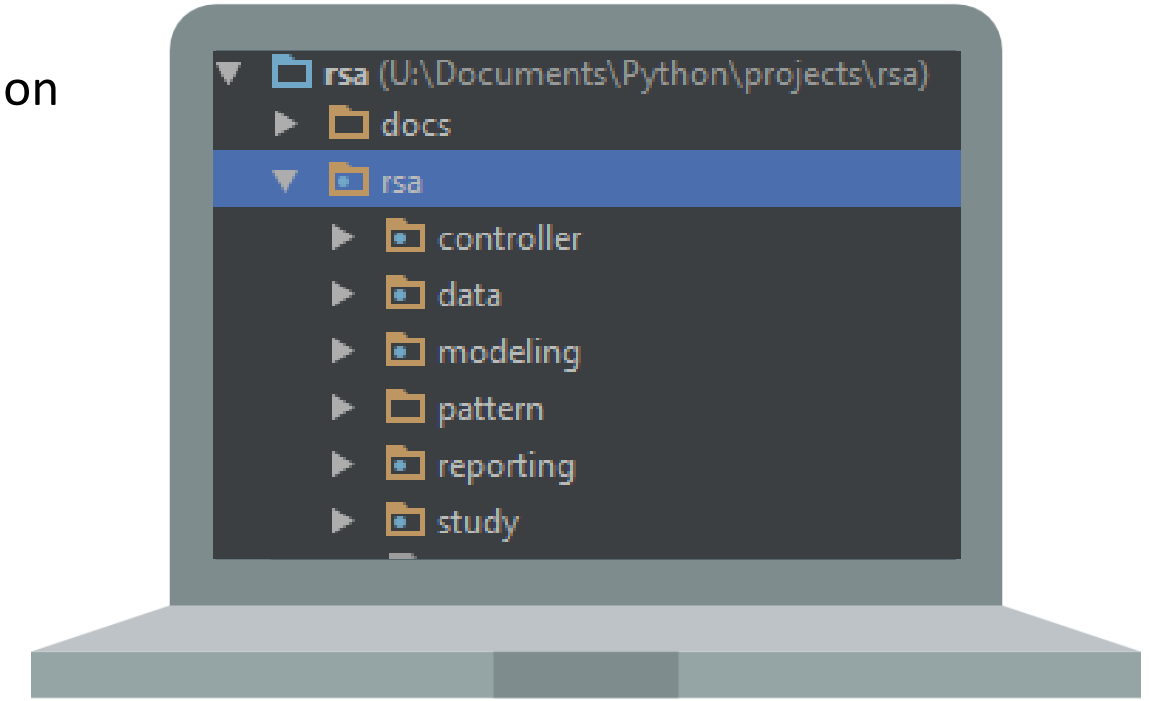
ANODE

Architecture: *Software*

Siemens PSSE v33
- For fine model building and manipulation

PowerGEM TARA
- For bulk computation, ORA protocols

ANODE Code Base (Python v2.7)
- For integration, orchestration, APIs



ANODE

Architecture: *API*

- Simple API for custom analysis underpins all system level activities

```
# create swing bus settings, voltage limit settings
study.create_settings()

# create load, temperature, outage schedules
study.create_schedules()

# here is where you would optionally change the load schedule
# Example:
study.load_schedule['07/17/18'] = 22500.0
study.load_schedule.save()

# here is where you would optionally change the temperature schedule
# Example:
study.temperature_schedule['07/17/18'] = 96.0
study.temperature_schedule.save()

# here is where you would optionally change the transmission outage schedule
# Example:
ticket = '12-34567'
start_date = study.start_date
end_date = study.end_date
study.outage_schedule.all_outages[ticket]['START_DATE'] = start_date
study.outage_schedule.all_outage[ticket]['END_DATE'] = end_date
study.outage_schedule.save()

# here is where you would optionally change the generator outage schedule
# Example:
index = 5
start_date = study.start_date
end_date = study.end_date
study.generator_outage_schedule.outages['data'][index]['StartDate'] = start_date
study.generator_outage_schedule.outages['data'][index]['StopDate'] = end_date
```

```
# here is where you would optionally select the seed case
from rsa.modeling.packages import ExportPackage
export = ExportPackage(r"""G:\ETReliabilityStudyAutomation\data\exports\my_export""")
study.import_seed_case(seed=export)

# optionally, you can omit the input and let the automation
# optimally select the seed case and copy it over to the study folder
study.import_seed_case()
study.create_seed_case_selection_report()

# create all in case from seed
study.create_all_in_case()

# create all rating set variations from all in case
study.create_alternate_limit_cases()

# create the tara outage files
study.create_tara_outage_file()

# create the files for tara AMB
study.create_tara_amb_files()

# run TARA AMB
study.run_tara_amb()

# create the files for TARA ORA
study.create_tara_ora_files()

# run TARA ORA
study.run_tara_ora()
```

ANODE

Architecture: *Automation*

- Many great Python libraries for task automation.
- To maximize simplicity, Windows Scheduled Tasks were selected.
- All top-level tasks follow the same pattern

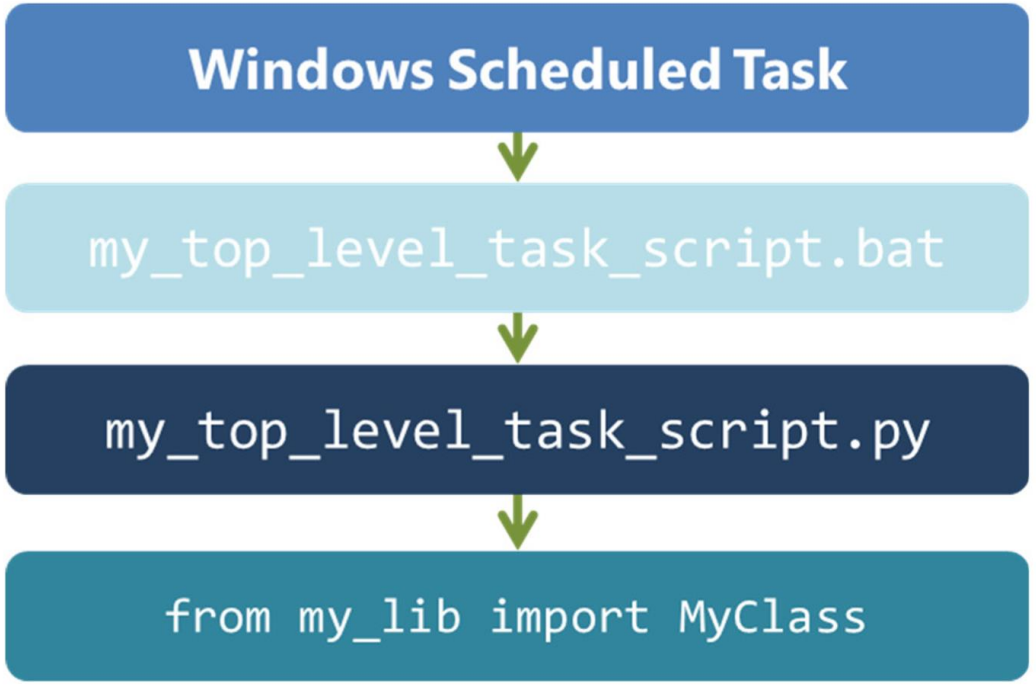


Figure 4: Automation Flow Chart

System Performance

Runtime/Run Count Comparisons

ANODE

System Performance: *Runtime Comparisons*

Table 2: Analysis Runtime Comparisons

	Manual Runtime	Platform Runtime	Rate Increase
Next Day Study	4-8 hours	15-20 minutes	**12-32X**
Ten Day Study	8-12 hours	15-20 minutes	**24-48X**
One Month Study	14-21 days	90 minutes	**224-336X**
Two Month Study	14-21 days	90 minutes	**224-336X**

ANODE

System Performance: *Run Count Comparisons*

Table 3: Calendar Day Coverage Comparisons

	Window Size *(days)*	**Manual Rate** *(runs per day)*	**Platform Rate** *(runs per day)*	**Manual Total** *(days per day)*	**Platform Total** *(days per day)*	**Rate Increase**
Next Day	1 day	1 per day	24 per day	1 day per day	24 days per day	**24X**
Ten Day	10 days	1 per day	24 per day	10 days per day	240 days per day	**24X**
One Month	30 days	1 per 30 days	1 per day	1 day per day	30 days per day	**30X**
Two Month	30 days	1 per 30 days	1 per day	1 day per day	30 days per day	**30X**
Total	71 days	N/A	N/A	13 days per day	324 days per day	**24.9X**

ANODE

Next Steps

- **Growing User Traction**
  - Improved Documentation
  - Feature Requests
- **Analysis Methodologies**
  - Study-on-Submittal
  - Optimal Outage Planning
  - Stochastic Analysis
  - Dynamic/Transient Stability Constraints for Outage Planning
  - Cascading Analysis
  - Synthetic Modeling
- **System Level Improvements**
  - Simple Web Front End
  - Data Source Improvements
  - General Refactoring
  - API Refinement

Thank You!

Kevin D. Jones, Ph.D.

kevin.d.jones@dominionenergy.com

304-767-4748

ANODE