Best Practices when Implementing Asset Analytics

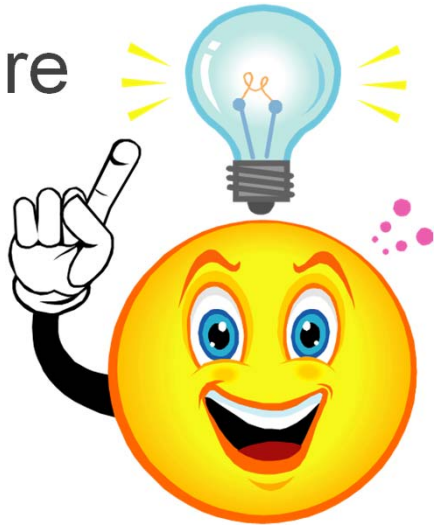
Stephen Kwan

Agenda

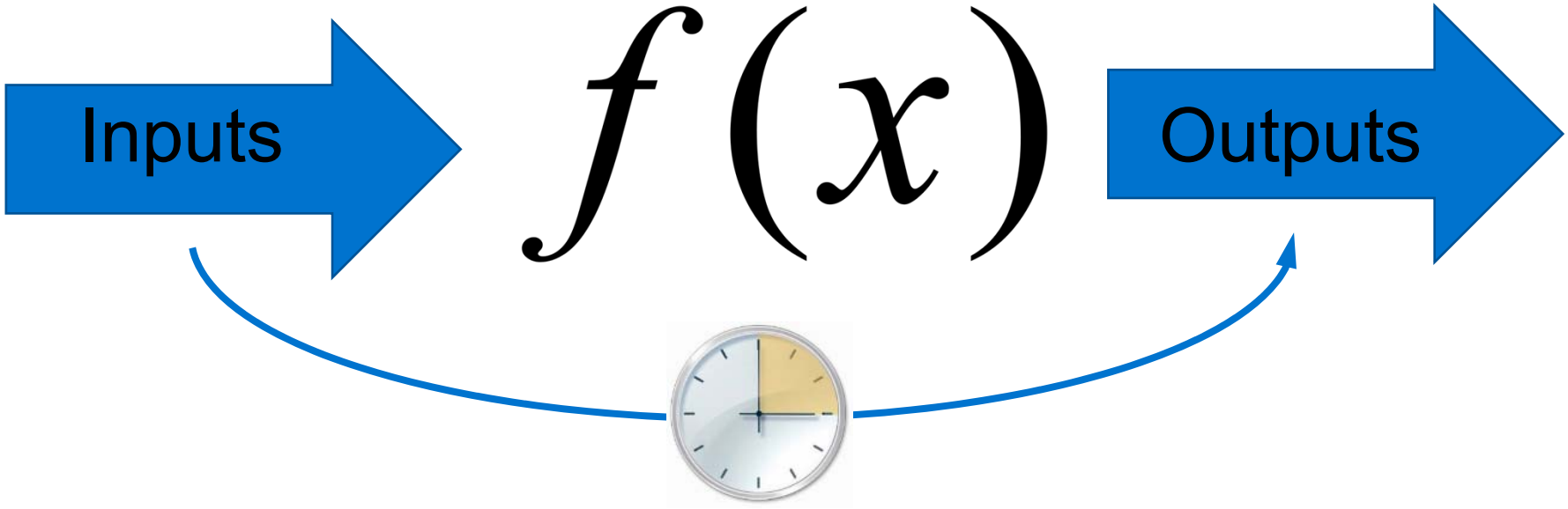
- Initial design principles
- Best practices
  - Planning
  - Tradeoffs – what to do, what not to do
  - Focus areas
  - Avoiding problems
- Summary

Initial Design Principles

- Leverage AF for context and event capture
- Easy to use
- Performance Equation function signature
- High performance
- 1 Analysis Service per 1 AF Server
- Auto backfills for restarts



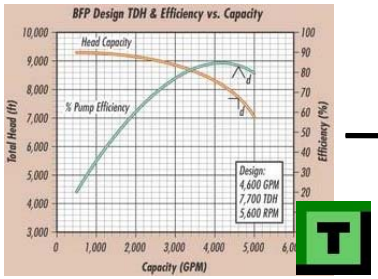
Basic Concept



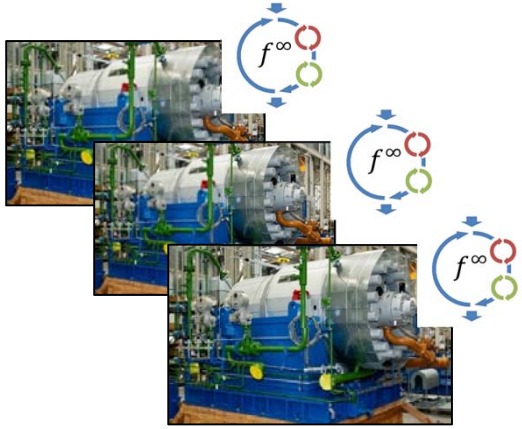
Workflow

Configure

Test

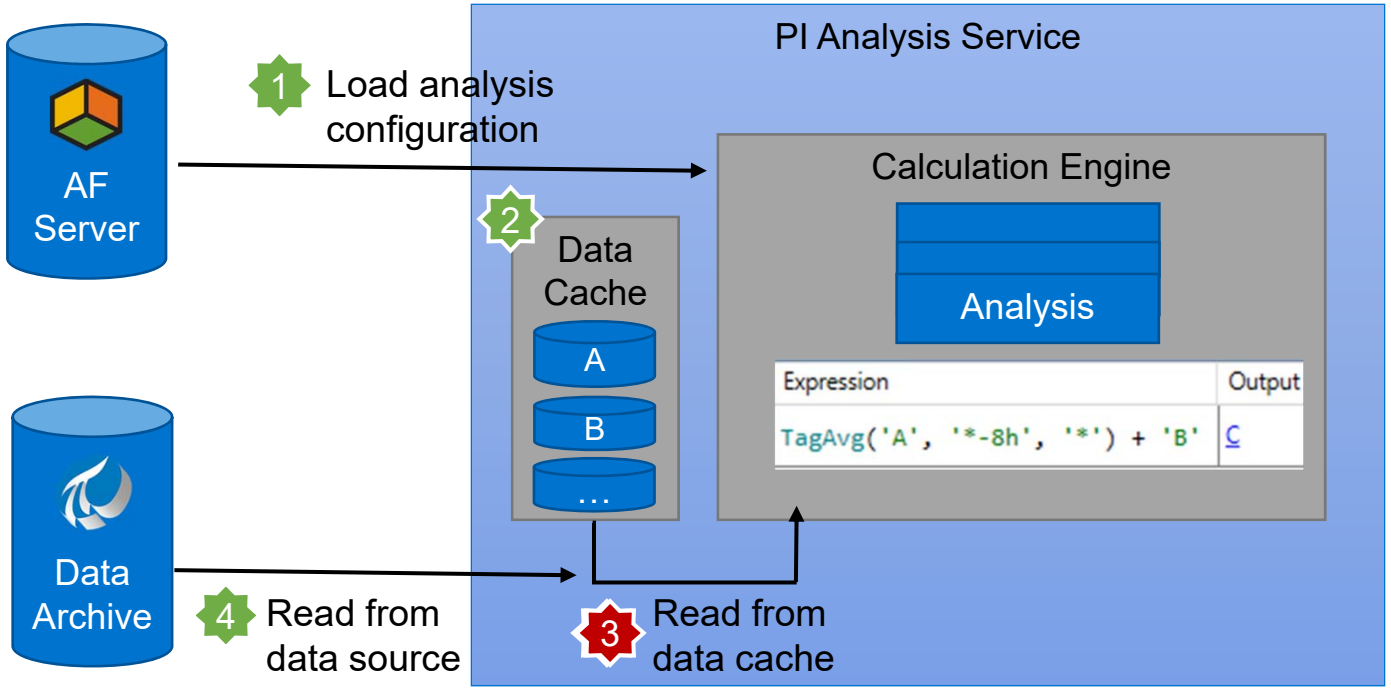


Schedule



Backfill/Recalculate

PI Analysis Service



Design Tradeoffs

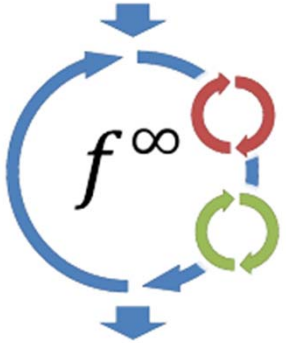
Optimized for

- Streaming analytics use case
- Prioritize real-time calculations over backfills/recalculations
- Leverage Data Cache
- Easy configuration

Not suitable for

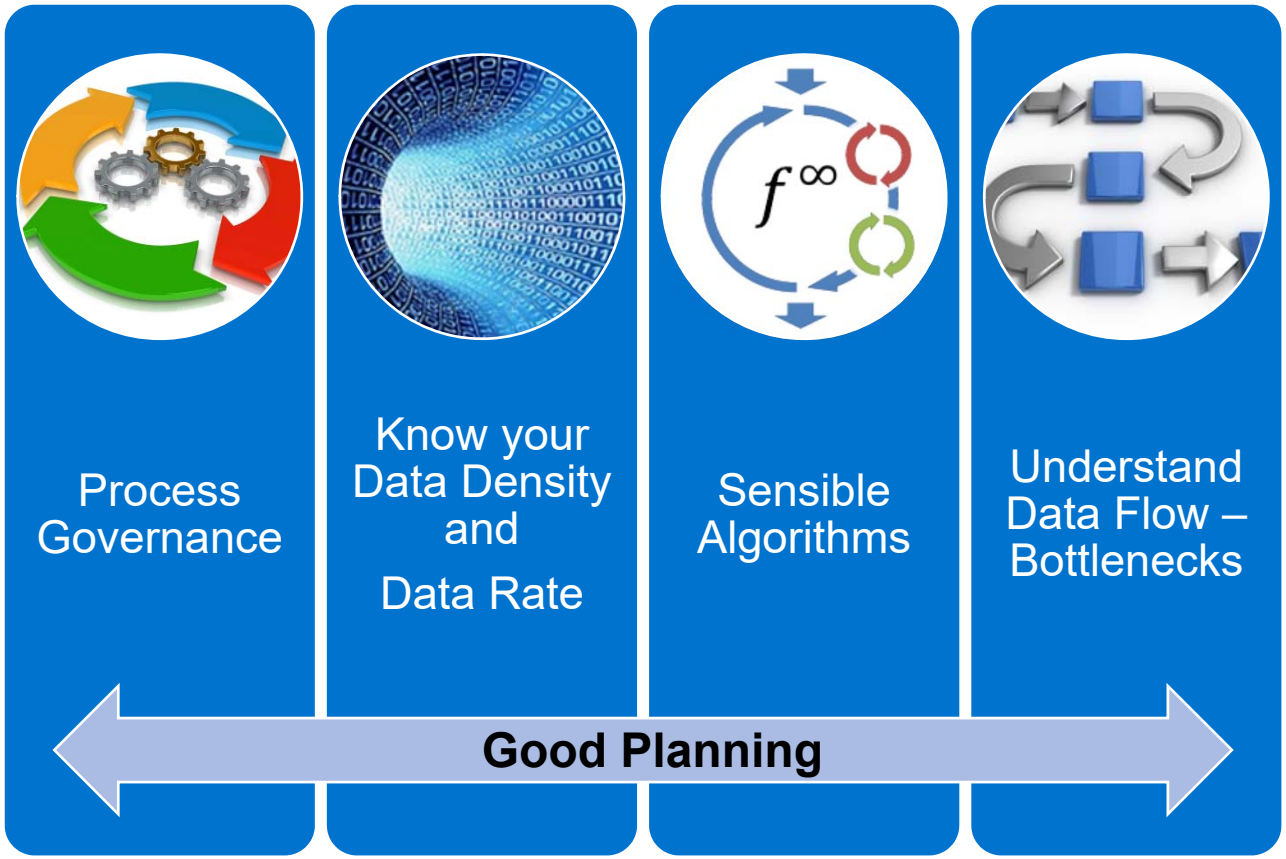
- Executing queries across really large number of attributes
- Extracting large amounts of time series data
- Ad-hoc calculations

Use cases: Streaming vs. Adhoc



Adhoc investigation and analytics development

Online, scheduled, streaming analytics



Change Management

- Consider setting up Dev or Test environment
  - Allows for experimentation while developing calculations
  - Many engineers could be writing calculations
  - Frequent changes can cause churn for the production system
  - Easier to isolate issues

Best Practices: Use Templates

- Provide manageability, consistency and governance
- Use templates for any repetitive work or for future extensions.
- A modification to the template is applied to all analyses from that template.
- Searching and filtering in UI is also easier with templates.
- Higher performance.

Best Practices: Configuration

- Retain tribal knowledge
  - Expressions should be written for readability
  - Add comments as appropriate
- Use PI Points for outputs – Higher efficiency and stores history
- Exit when you're done – don't continue to run calculations
- Use Categories for filtering

Understand the Impact of your Analyses

- Some functions such as summary functions (TagAvg etc.) might require a lot of data to perform a calculation on the client side
  - Use a smaller range or evaluating less often if the range is larger
- Hit the cache and not I/O to Data Archive
- Do not trigger faster than evaluation rate
- Watch out for dependent analyses
- Plan for out of order data – recalculation

Avoid Common Bottlenecks

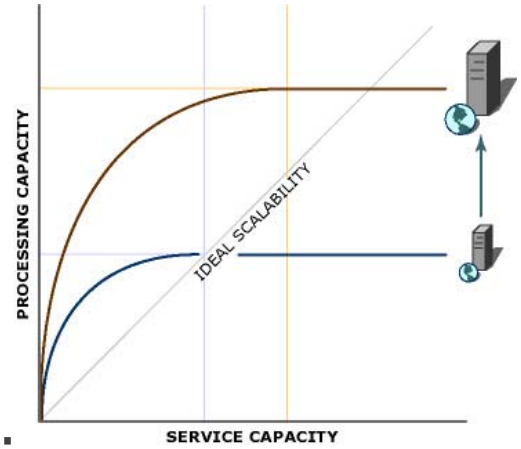
- Inputs from slow data references, e.g. Table Lookup
- Inputs requiring round trip to Data Archive (I/O bound)
- Too many analyses triggering at the same time
- Sudden slug of triggering inputs

High Availability

- PI Analysis Service supports failover using Windows Server Failover Clustering (WSFC)
- Use PI Buffer Subsystem for writing PI Point outputs

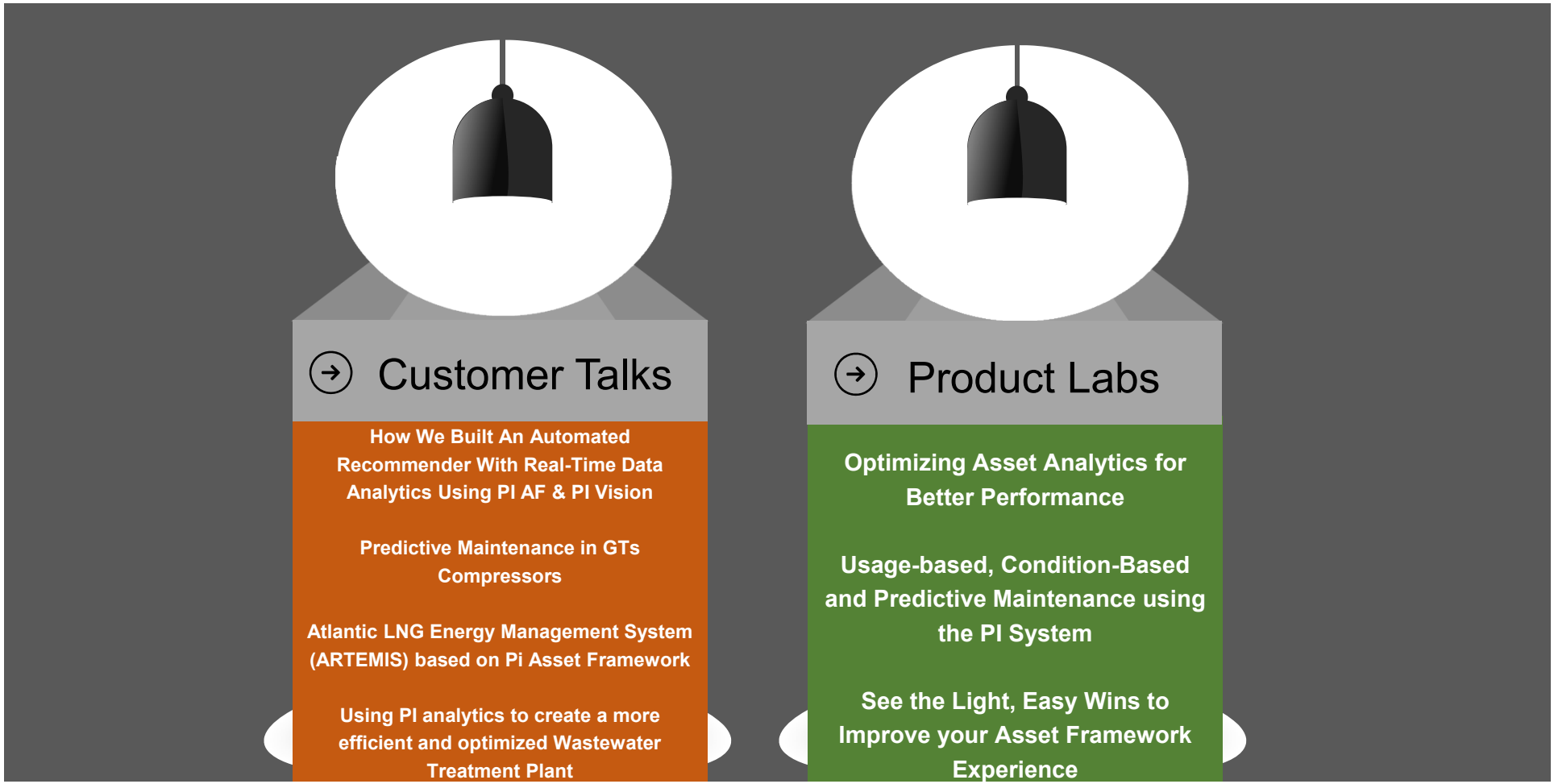
Lastly

- Nothing scales forever
  - Use multiple systems if needed
- Optimizing single system has limitations
  - Hard to recover from a perturbation



Summary

- Governance – Test + production system
- Sensible data density and data rate
- Streaming vs. adhoc calculations
- Only calculate what's needed
- Avoid round trips to Data Archive for inputs
- Highly tuned system may be susceptible to small perturbations



Communicate with OSIsoft Product Managers

https://feedback.osisoft.com

If it is not shared on the feedback portal, it didn't happen!

- **Stephen Kwan**
- skwan@osisoft.com
- Product Manager
- OSIsoft, LLC

Questions?

Please wait for the **microphone**

State your **name & company**

Please remember to…

Complete Survey!

Navigate to this session in mobile agenda for survey