

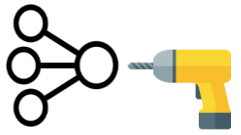
Data Diviner

CI DataMart for Automation

Server System Compliance

Daily Check Automation

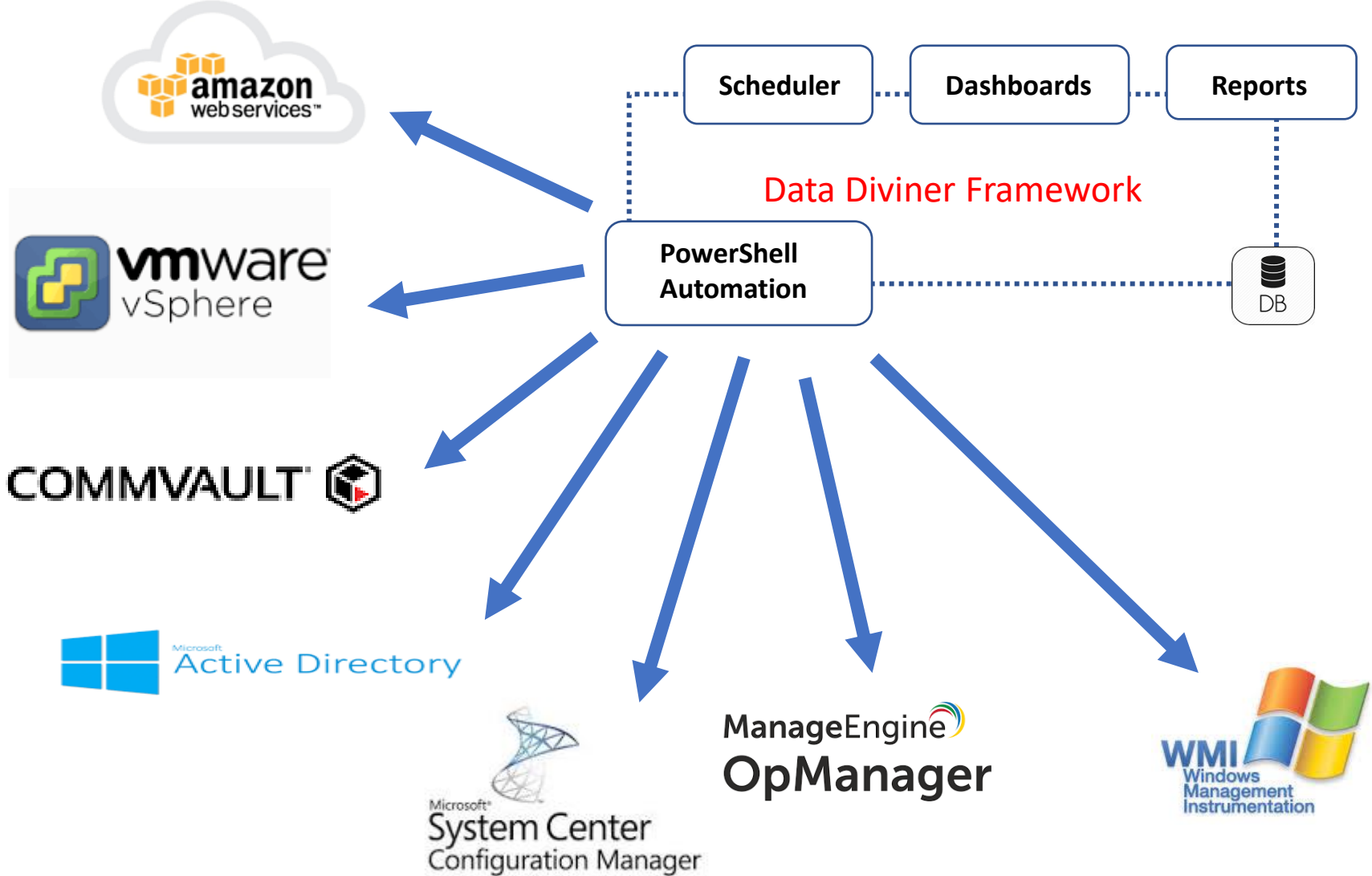
Patching Automation



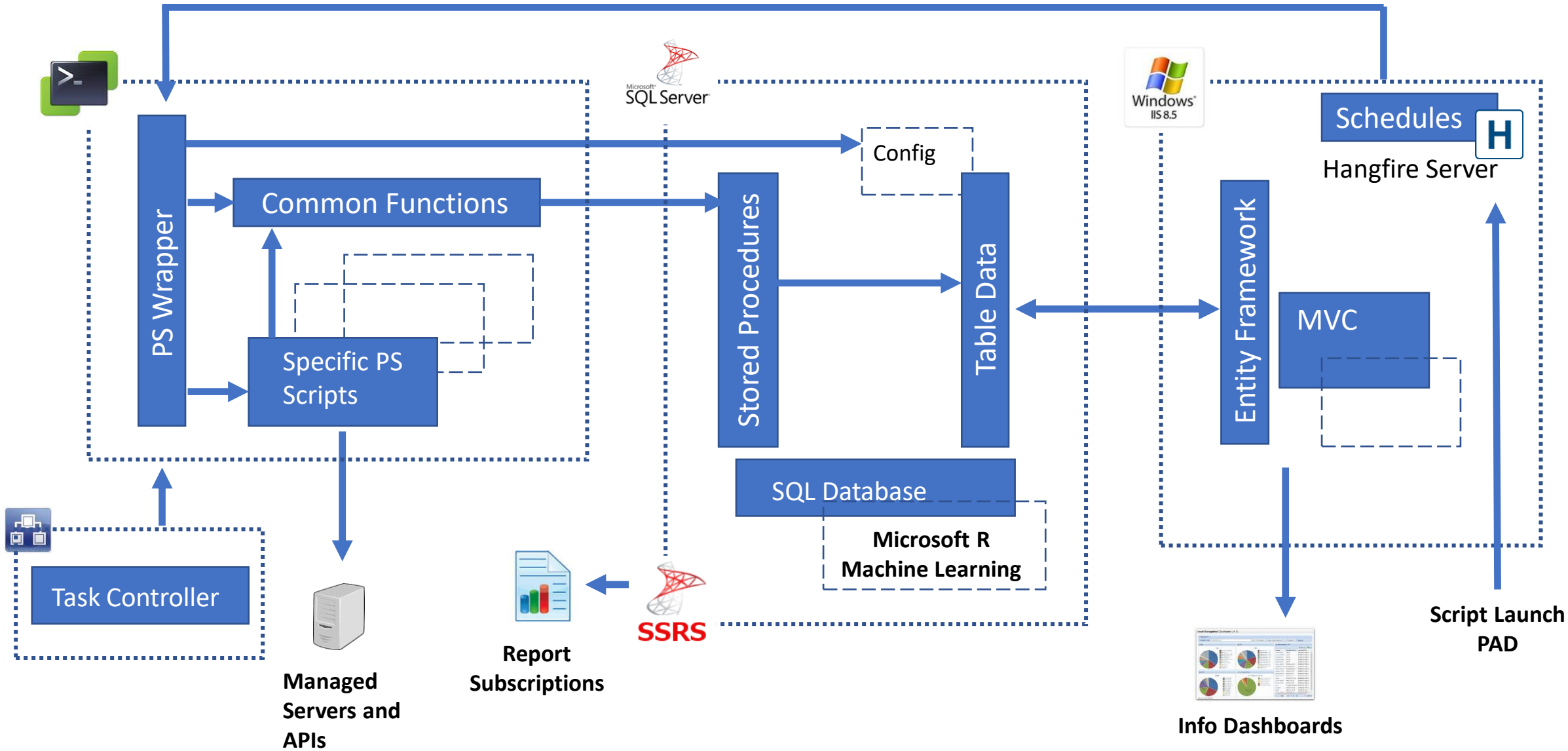
Data Diviner (Benefits)

- Visualize your Server Infrastructure in one place
- Aggregate Core Systems Status
- Execute Automation Tasks with PowerShell
- Report across all captured System Data
- Automate Health and Daily Checks
- Patching\Hotfix Automation
- Agent Compliance (are devices configured for core infrastructure)
- Machine Learning across Key Information (Coming Soon)

Data Diviner (Overview)



Infrastructure Automation & Reporting Framework



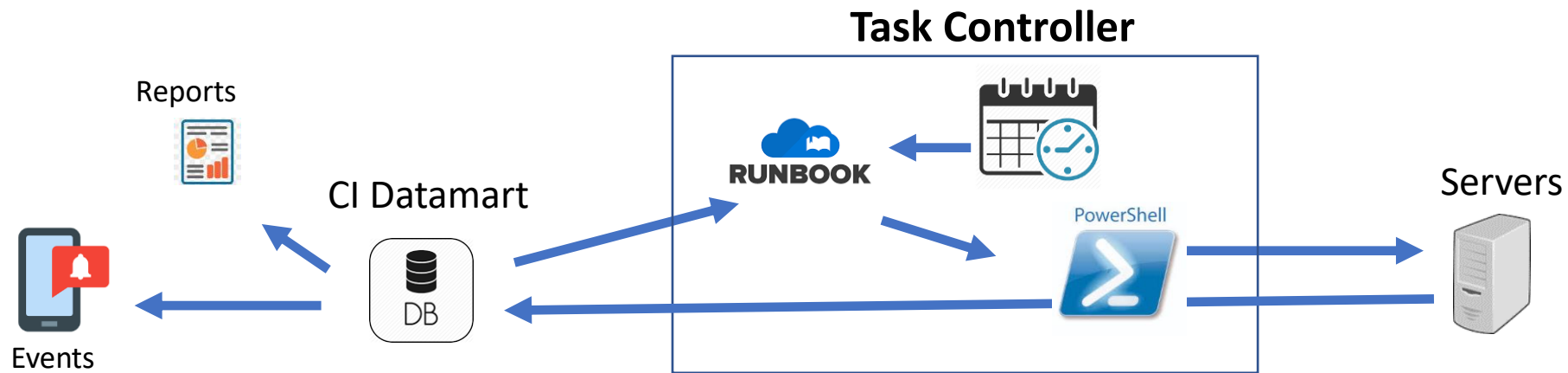
Main Achievements

- PowerShell Framework Integration with MVC
- Scheduling Engine
- Commvault API and OPManager API Methods
- vSphere Collector Methods
- AWS Collector Methods
- SCCM Methods – Client and Server
- WMI Collector Methods – Machine and Software
- Basic SCCM Patch Sequencer
- Initial Deployment (Server 2016 and SQL 2017)
- Document some core concepts

Task Scheduling and Control

Option 1

System Centre Orchestrator could become the central control of all Automation tasks
Schedules would be in SCORCH

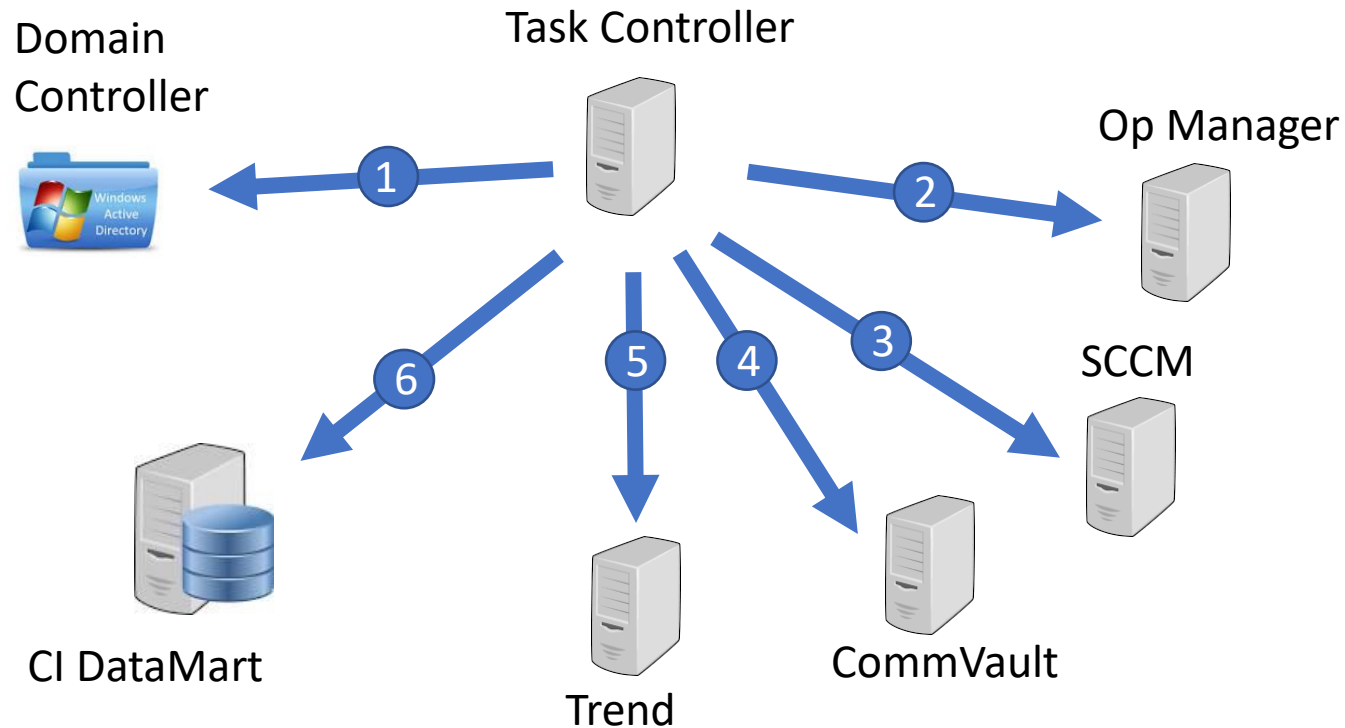


Option 2

Job Scheduling Plugin for MVC Hangfire (See Hangfire.io)

- Configuration data would exist in the CI Datamart for both cases
- All Data would be posted into CI Datamart for both cases

Populating CI DataMart (Method 1)



Server Discovery

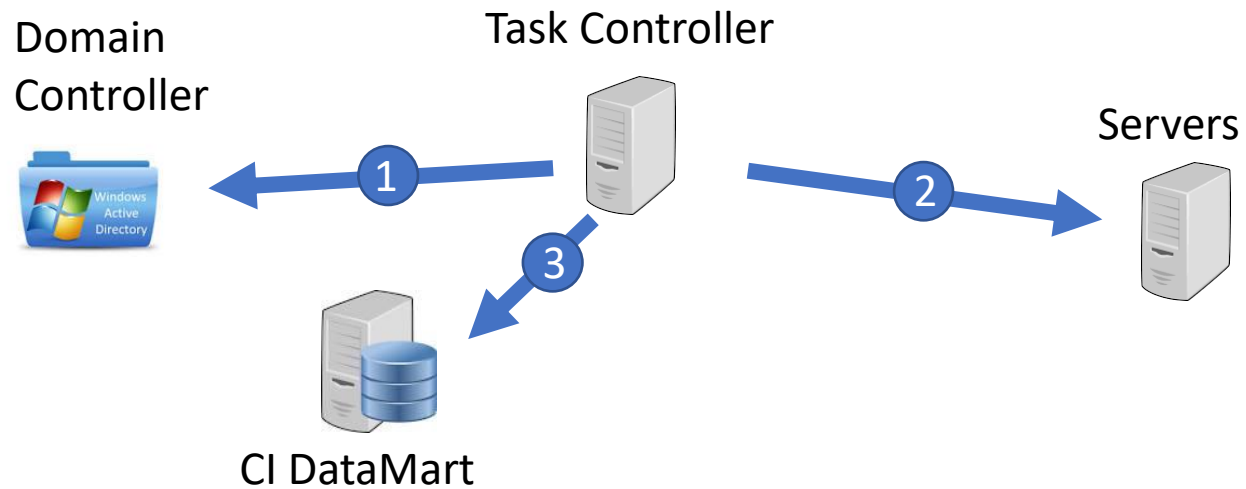
- 1 PowerShell Get Servers
- 2 Query API Get Devices
- 3 WMI Get Summaries
- 4 API to get agents
- 5 Database Query / Web Scrape
- 6 Store Discovered Data

This method we collect data from the Systems that manage the end points directly using either PowerShell, REST API, Database queries or Web Scraping.

Disadvantage: API Security, Many different Methods, Limited Endpoint Information

Advantage: Overall System Summary from Single Point good for Daily Checks

Populating CI DataMart (Method 2)



Server Discovery

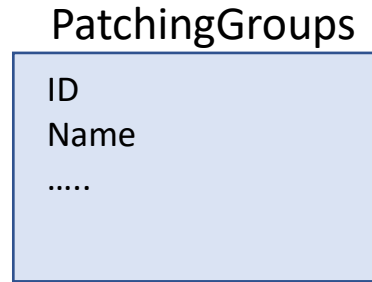
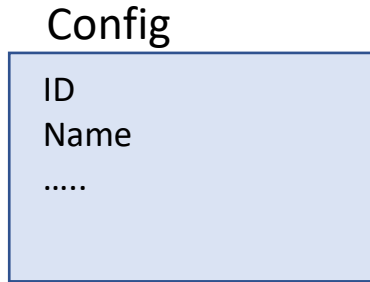
- 1 PowerShell Get Servers
- 2 PowerShell WMI Queries
- 3 Store Discovered Data

This method we collect data from the Server end points directly using PowerShell as core scripting mechanism but would utilize WMI and Registry Configuration Items

Disadvantage: Lots of Endpoints to Communicate, Network Security ACLs

Advantage: System Info (As Built), Consistent Scripts all Machines

CI Datamart Tables



Tables used for config and special task control and sequences
Can contain the Specific Application Checks

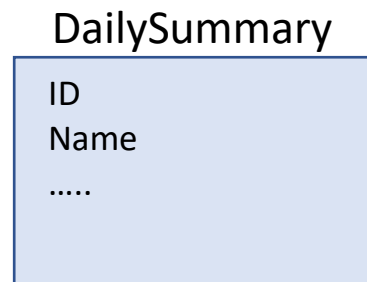
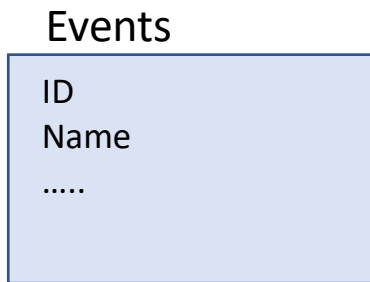
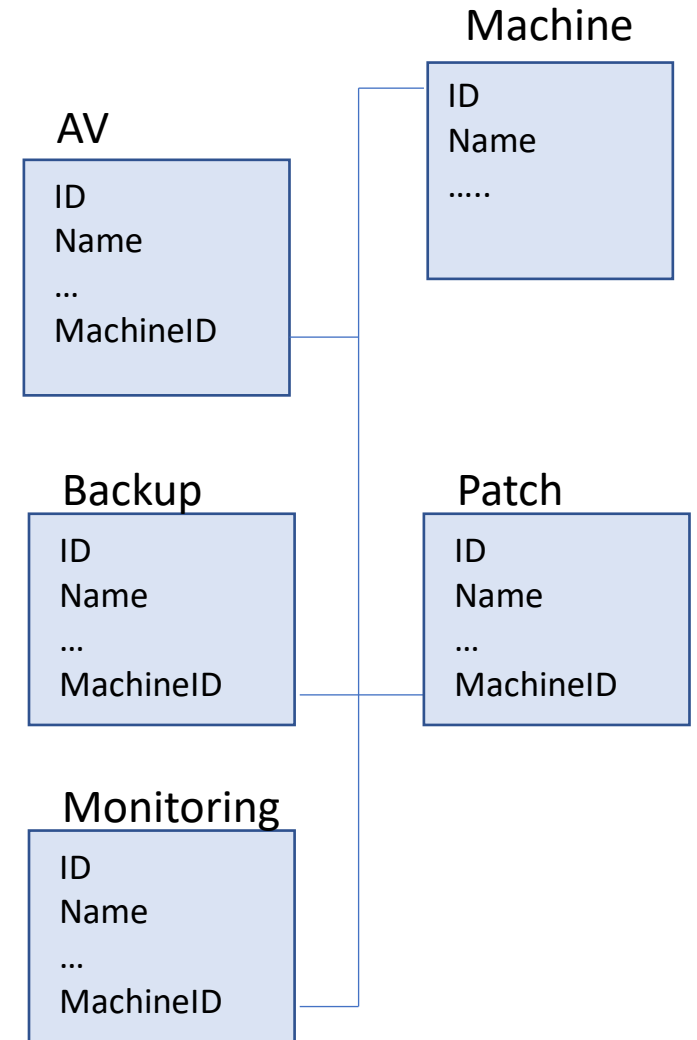
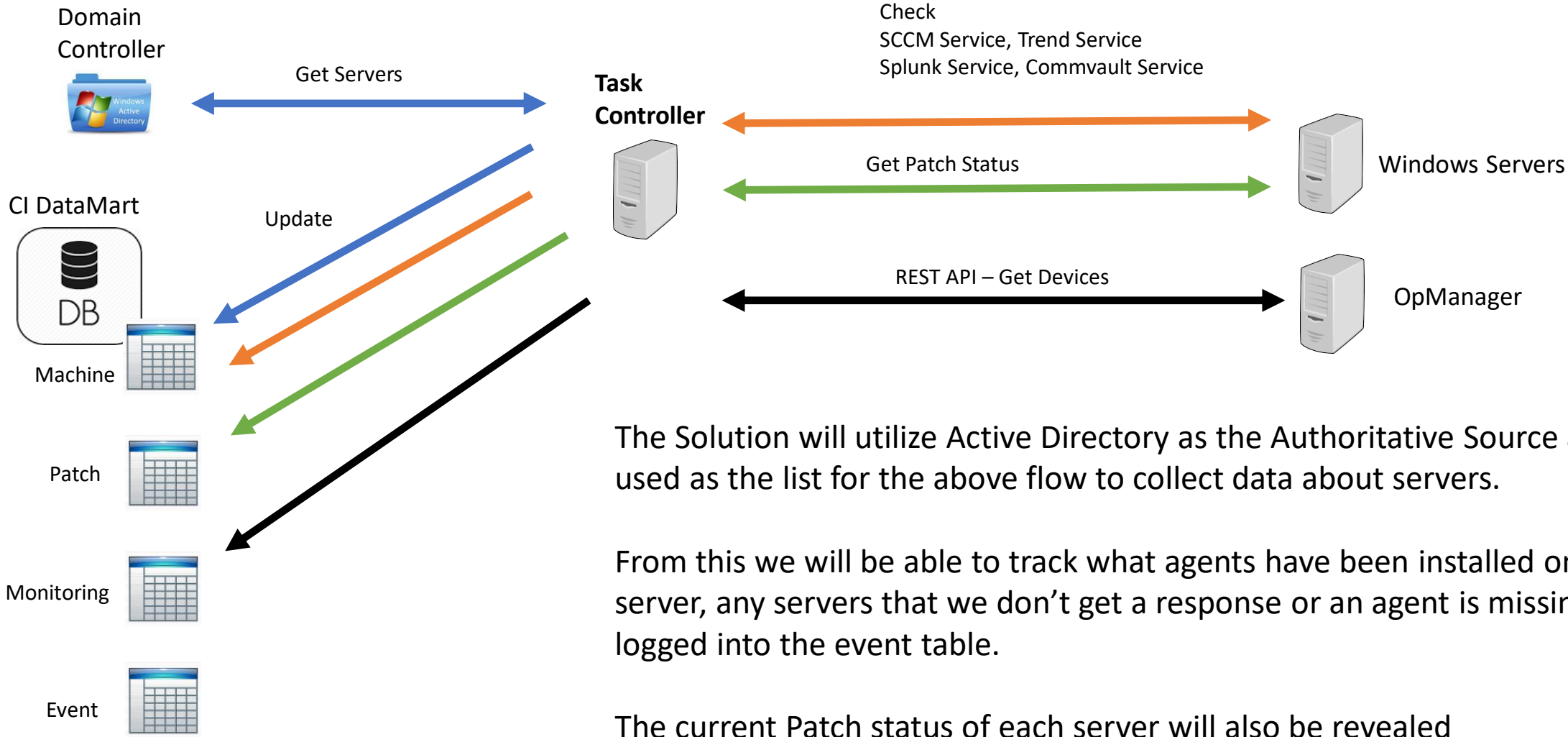


Table used for capturing what is happening with data consolidation



Tables with raw system data

Populating CI DataMart (Base Data)



The Solution will utilize Active Directory as the Authoritative Source and will be used as the list for the above flow to collect data about servers.

From this we will be able to track what agents have been installed onto each server, any servers that we don't get a response or an agent is missing will be logged into the event table.

The current Patch status of each server will also be revealed

Server to System Remediation

Ensuring all Servers are added to critical Infrastructure Service Endpoints

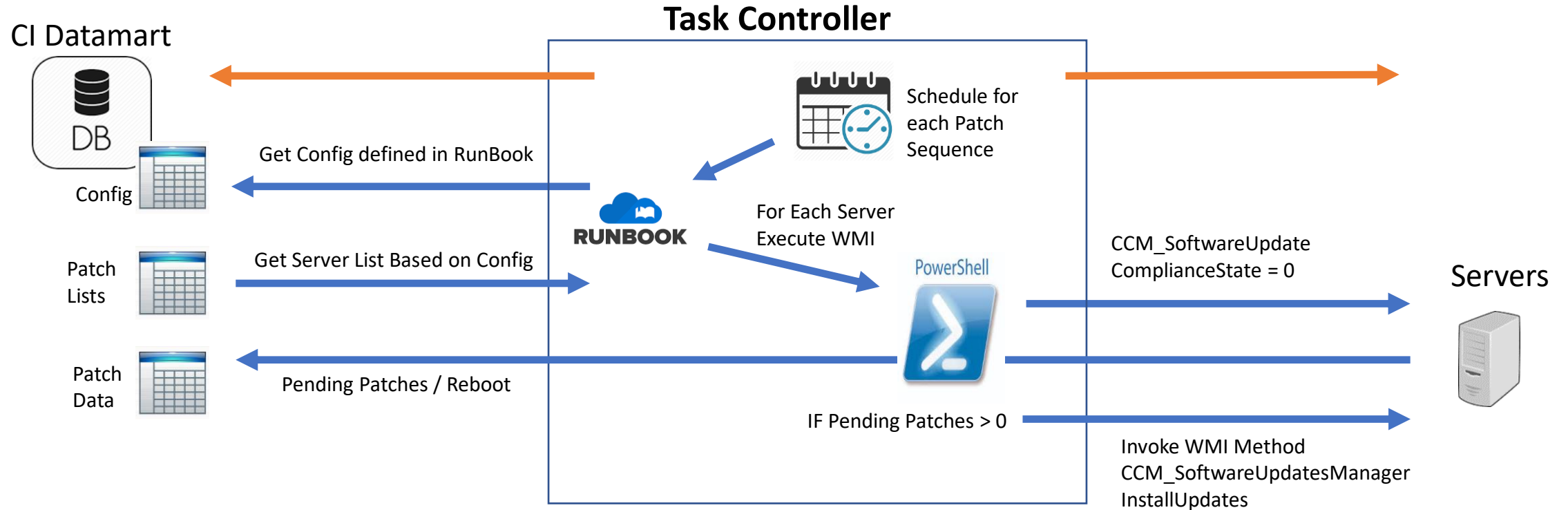
- Once each of the systems have been queried and tables populated, Machine Table will contain all discovered servers and which system they reside.

ID	ADName	DNSName	OS	OSVersion	ADDescription	HasMon	HasSCCM	HasBU	HasAV
1	MRP-TID-D03	MRP-TID-D03.mrp.net.nz	Windows Server 2012 R2	6.3 (9600)	TID Dev SQL	0	0	1	0

- By querying Servers directly to collect agent availability allows us to extend the data captured
- WMI provides detailed System Information the following could be added to the Database
CPU, Disk, Memory, Services, Installed Apps....
- We could run these checks against all servers weekly any issues will be logged to the event table and we could raise a notification for investigation
- Hypervisor API ie PowerCLI vSphere info on windows and Non Windows machines
- Data in the database could be accessed by SSRS reports or an MVC Dashboard / Views

Patching Automation

← Pre Job



- Task Controller controls the deployment either scheduled or manually execute
- Runbook determines the Server List to be executed based on Config entry in DB
- Server Patch Lists defined in the DB Table
- Server Patch Compliance can be checked at any time (could be used to trigger which machines)
- Assumes SCCM have patches configured for machines and are available

Daily Check Automation

A Task Controller

- Can execute AD Checks (PowerShell, DCDiag)
- Call REST API eg Commvault or OpManager Status information
- Can Execute vCentre, Exchange PowerShell scripts
- Can Execute XenApp SDK PowerShell Cmdlet's
- Can Execute PowerShell to get SCCM Status
- Execute Custom .NET Scripts to check Logs or Databases

The Configuration for each of these query results would be stored in the Database

The Data can be accessed using

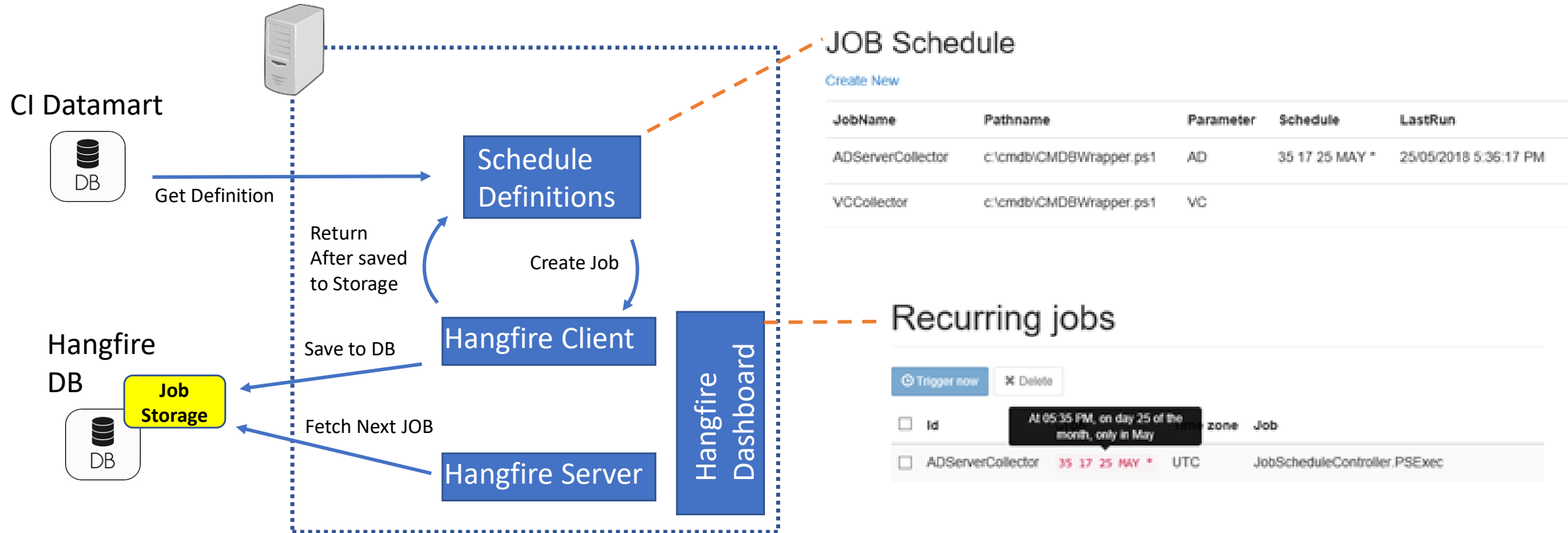
- A Simple Web Site to provide dashboards and views
- And SSRS to have standard reports

These checks need to be standardised so they can be used across any Customers

Solution Requirements

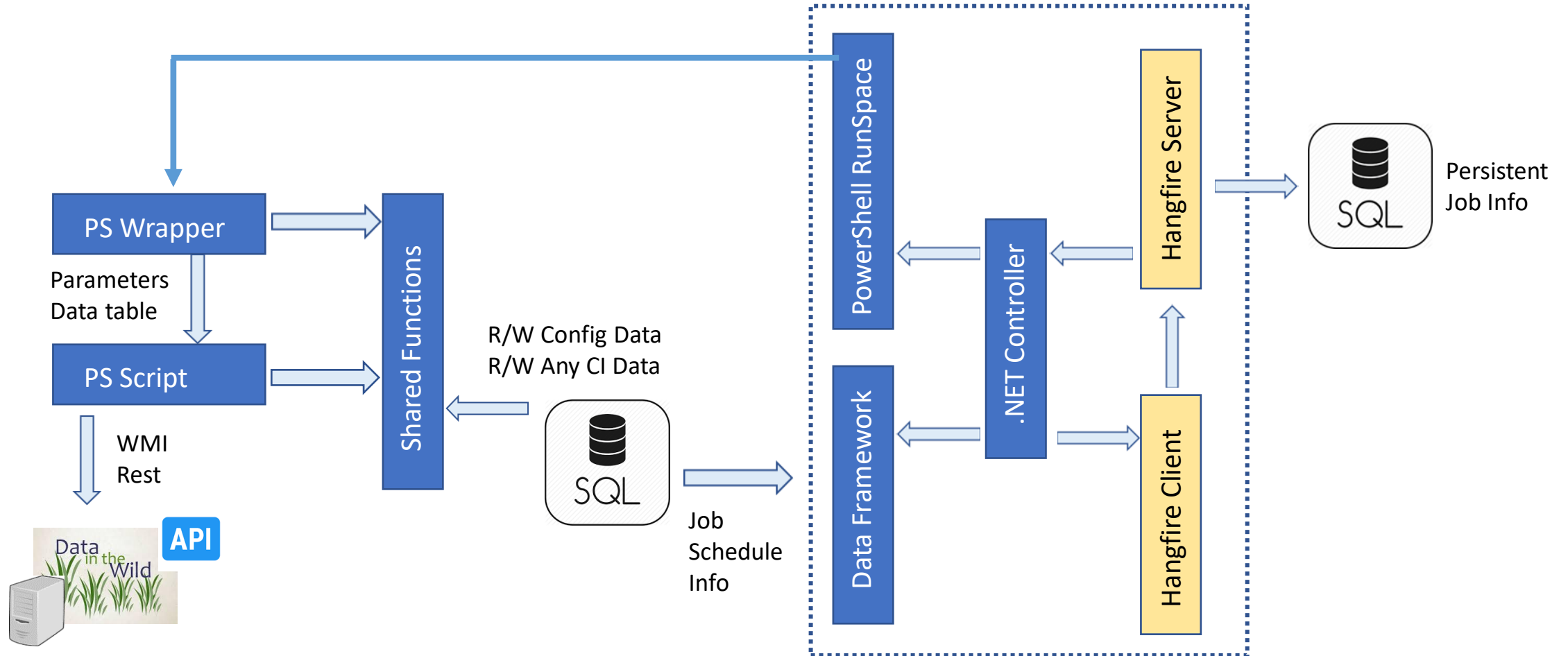
- MS SQL Server 2017 with SSRS
- IIS 8 for MVC Web App
- PS Wrapper and Scheduler
- AD, Exchange, XenApp Management Packs
- VMware PowerCLI

Hangfire Scheduler – MVC Nuget Package

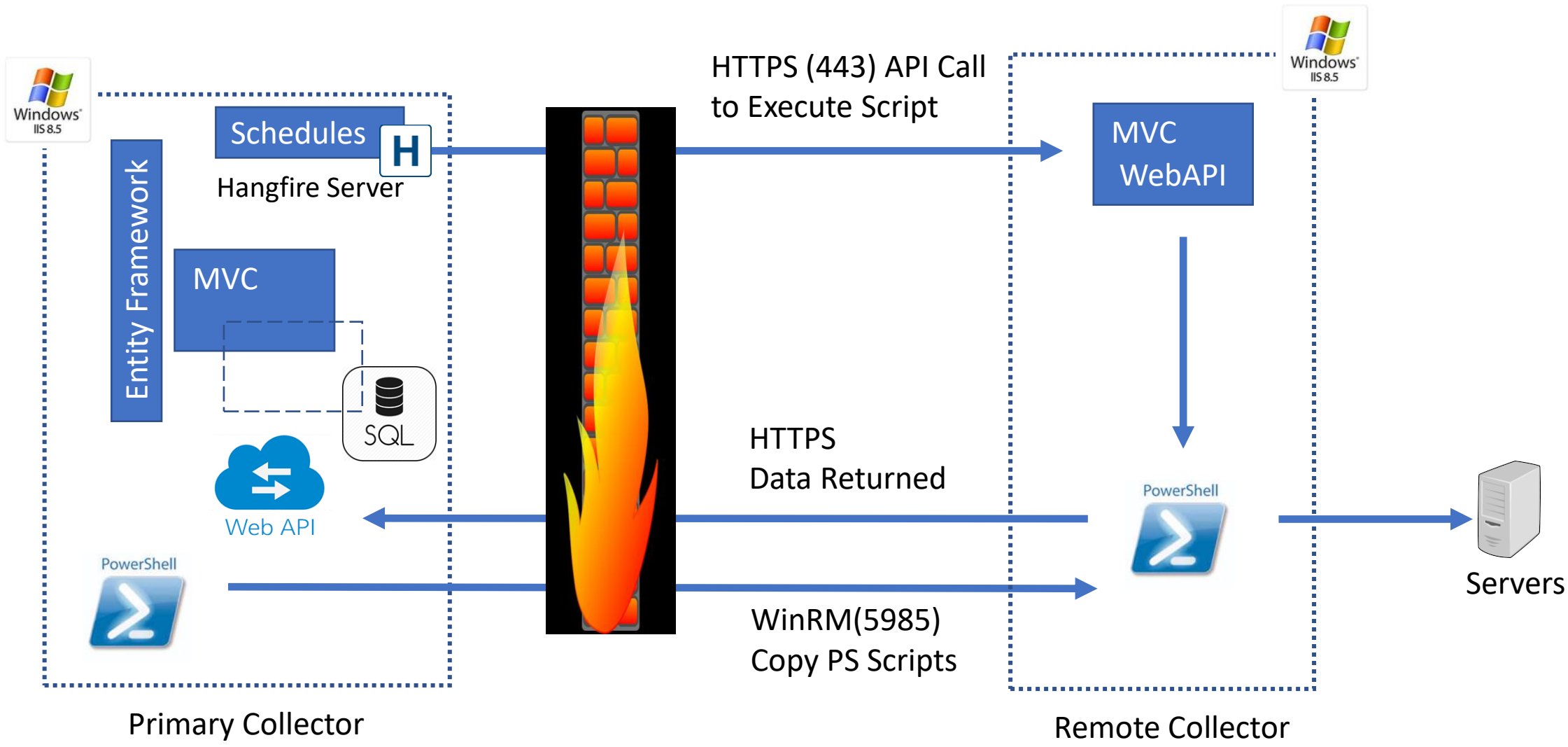


- MVC Task submitted to Hangfire to manage
- Dedicated Dashboard for Submitted Jobs
- All Jobs Stored in Database

Scheduling and PowerShell



T2 Remote Collector



Ideation

- Create Alerts when new machine appears
- Create Alerts when machine doesn't meet SOE agents/services
- Create Alerts around windows errors and Uptime >
- Create Alerts when new software Installed
- Create Alerts when SCCM client has issues
- Create Alerts about VMware DC errors and VMTools
- Create change based on upcoming Automation Schedule
- Licensing changes , new Software installed
- Add Manual Form for External - non Integrated systems
- Machine Learning around disk space and Backup space

Backlog and Bugs

- Add EICAR AV Test
- AWS Collector
- Full Workflow engine.
- SQL Databases collector
- Machine Learning using R
- Feed Data to NoSql
- History Functions
- Create WinRM channels methods
- Create Remote Collector (HTTPs)
- Script Execution Failure error codes
- Add Job ID to Script execution and to Collector Messages.
- Pagination Clean up
- Commvault JobSummary
- VM Vsphere is deleted ?
- SCCM Component status order
- SCCM site status order
- Reorg Windows Data Summary Charts
- Config summary no response to ping - WMI...
- Secure Config and Jobs page
- User data have group membership , which machines they logged in from
- SVC Checker new add auto
- Second VC