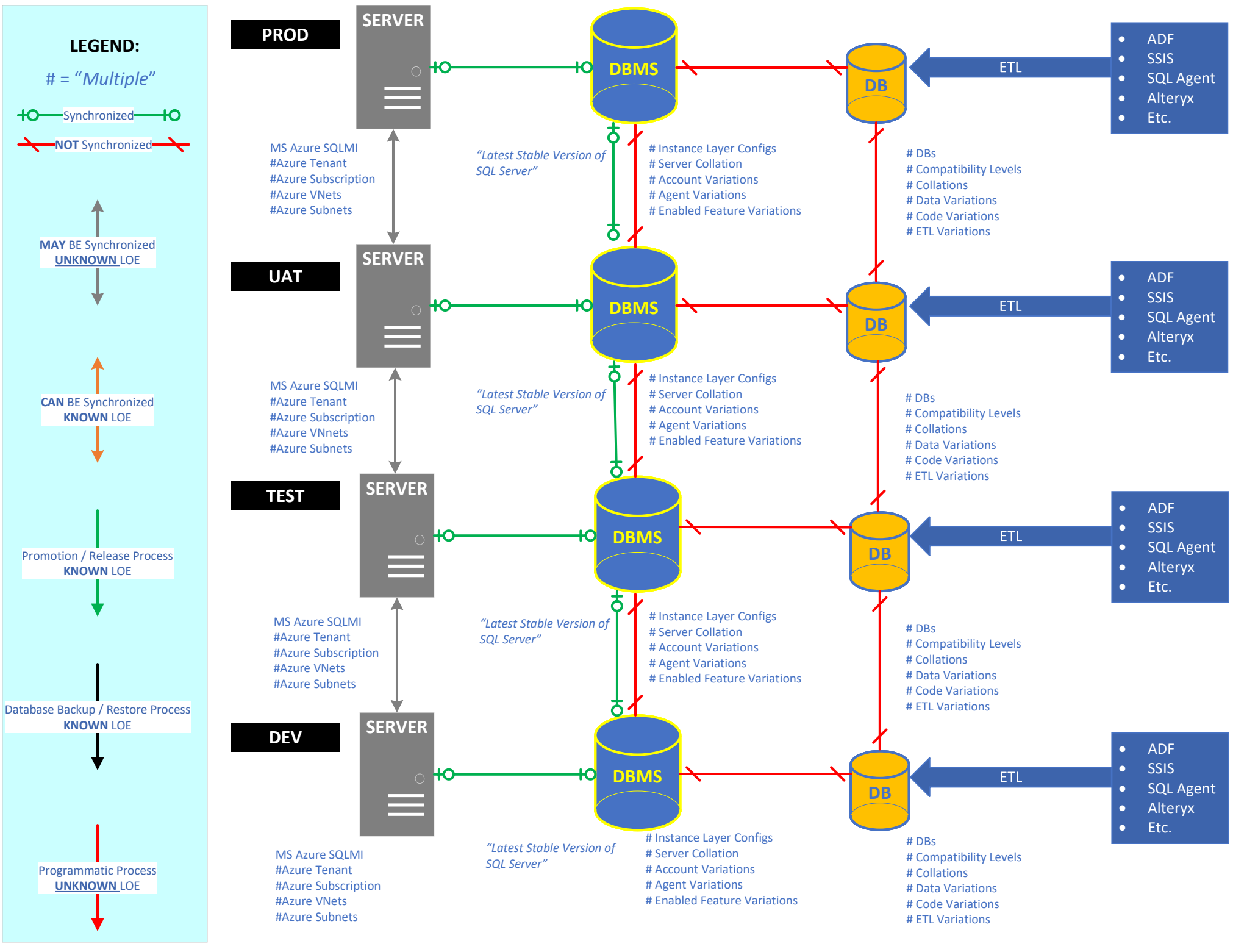
CURRENT STATE: 4 Distinct environments, serving the same solution (Comprehensive EDW; "*Rogue1*"), with virtually no interoperability, very few (informal) standards in place, and numerous configuration inconsistencies

- Code (DML, DDL, logic) deployed directly to PROD ("*Doing development in Prod.*")
- Real-world insights derived from DEV ("*Treating Dev. as if it were Prod.*")
- ETL routines into lower environments – each routine curated independently
- Configuration facets dictated from SNC architecture standards ("*VERY limited IPs in the Rogue1 subnets – tech debt from SNC" - per Al Dass*)
- Configuration facets inconsistently applied across SDLC environment
- Numerous server level collation levels
- Numerous DB compatibility levels
- No formal process to enable effective development or testing
- No formal process to promote from lower environments to "PROD"
- No formal process to obfuscate / mask sensitive data attributes
- No informal development / prototyping environment



OBJECTIVE: To develop a formalized process to logically refresh EDW data and schemas to facilitate effective development and testing, using a sequenced, cascading approach in consideration of current overall environment state.

This proposed process is in strict consideration of a (one-time) baseline setting project for the production environment as a prerequisite

- Set baseline for PROD:
  Copy (**backup / restore**) to UAT
  Copy (**backup / restore**) to TEST
  Copy (**backup / restore**) to DEV

- Begin Scheduled / Automated "Copy Back" Process that **backs up each database**, and **restores each database** that exists in the PROD environment, into the UAT environment. (*Interval TBD*)

- **Data** in PROD is populated through **ETL processes** using v**arious products and techniques** (*ADF*, *SSIS*, *SQL Agent*, *Alteryx*, *Etc*.)

- Engineer / Adopt process that promotes **DDL** changes that occur in DEV environment to TEST environment using **GitHub**. (*Interval TBD*)
- Engineer / Adopt process that promotes **DML** changes that occur in DEV environment to TEST environment using **GitHub**. (*Interval TBD*)

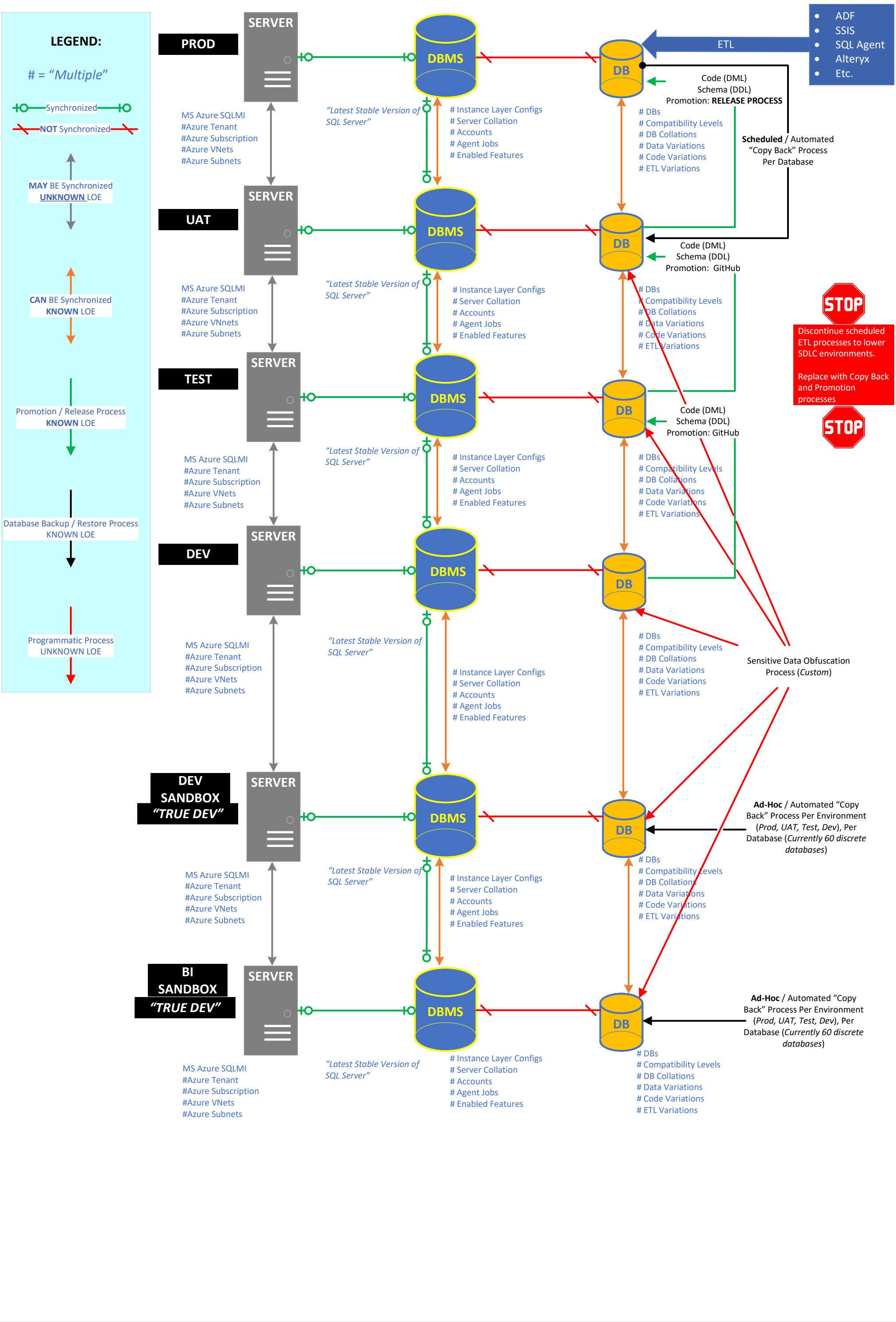
- Engineer / Adopt process that promotes **DDL** changes that occur in TEST environment to UAT environment using **GitHub**. (*Interval TBD*)
- Engineer / Adopt process that promotes **DML** changes that occur in TEST environment to UAT environment using **GitHub**. (*Interval TBD*)

- Engineer / Adopt process that promotes **DDL** changes that occur in UAT environment to PROD environment using manual **Release Process**. (*Interval TBD*)
- Engineer / Adopt process that promotes **DML** changes that occur in UAT environment to PROD environment using manual **Release Process**. (*Interval TBD*)

- Engineer / Adopt process that sequentially refreshes **DATA** in UAT via manual / automated "Copy Back" process from PROD and subsequently promotes **DDL / DML** changes to UAT from TEST using GitHub. This will facilitate an integrated testing environment (*i.e. fresh data, and latest code versions.*)

- Make available Sandbox environment that enables development / prototyping of **DDL** changes against databases that are individually backed up, and restored from higher environment(s) using **Ad-Hoc "Copy Back" Processes.** Depending upon use-case, these DDL changes are promoted to (*formal*) DEV environment and included in the SDLC.

- Make available Sandbox environment that enables development / prototyping of **DML** changes against databases that are individually backed up, and restored from higher environment(s) using **Ad-Hoc "Copy Back" Processes.** Depending upon use-case, these DDL changes are promoted to (*formal*) DEV environment and included in the SDLC.



OBJECTIVE: To develop a formalized process to logically refresh EDW data and schemas to facilitate effective development and testing, using a sequenced, cascading approach in consideration of current overall environment state.

This proposed process is in strict consideration of a (one-time) baseline setting project for the production environment as a prerequisite

- Set baseline for PROD:
  Copy (**backup / restore**) to NON-PROD / INTEGRATION
  Copy (**backup / restore**) to SANDBOX1
  Copy (**backup / restore**) to SANDBOX2

- **Data** in PROD is populated through **ETL processes** using **various products and techniques** (*ADF*, *SSIS*, *SQL Agent*, *Alteryx*, *Etc*.)

- Engineer / Adopt process that promotes **DDL** code that is created / changed in SANDBOX1 environment to NON-PROD / INTEGRATION environment using **GitHub**. (SCHEDULED OR MANUAL *Interval TBD*)

- Engineer / Adopt process that promotes **DML** code that is created / changed in SANDBOX1 environment to NON-PROD / INTEGRATION environment using **GitHub**. (SCHEDULED OR MANUAL *Interval TBD*)

- Engineer / Adopt process that promotes **DDL** code that is created / changed in SANDBOX2 environment to NON-PROD / INTEGRATION environment using **GitHub**. (SCHEDULED OR MANUAL *Interval TBD*)

- Engineer / Adopt process that promotes **DML** code that is created / changed in SANDBOX2 environment to NON-PROD / INTEGRATION environment using **GitHub**. (SCHEDULED OR MANUAL *Interval TBD*)

- Engineer / Adopt process that promotes **DDL** code that is created / changed in NON-PROD / INTEGRATION environment to PROD environment using manual **Release Process**. (MANUAL (controlled) *Interval TBD – (Weekly release window)*)

- Engineer / Adopt process that promotes **DML** code that is created / changed in NON-PROD / INTEGRATION environment to PROD environment using manual **Release Process**. (MANUAL (controlled) *Interval TBD – (Weekly release window)*)

- Engineer / Adopt processes that refresh lower SDLC environments with **DATA** from PROD via either manual or automated "Copy Back" process and subsequently promotes **DDL / DML** to NON-PROD / INTEGRATION environment from SANDBOX environments using GitHub. This will facilitate an integrated testing environment (*i.e. fresh data, which is then overlaid with latest code versions.*)