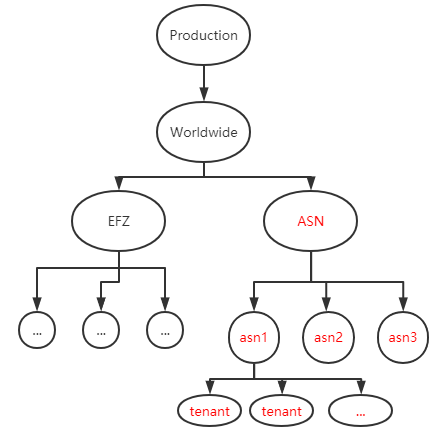
Tenant-Asn Mapping

**Overview**

Currently, SHD posting for networking scenario only supports to post as EFZ level. As the enrichment of alerting scope in DAREv3, the requirement of SHD is to support more posting scopes as well. ASN(+Metro) is one of the scopes need to proceed.

The Tenant-Asn Mapping will be generated from daily café log, it would be updated into SHD DB weekly. We mainly focus on the **TOP 200 ASNs(corresponding to DAREv3 alerting ASNs)**, and get all important tenants accessing Microsoft through these ASNs. Based on this prepared mapping, SHD posting can navigate to notify these tenants accurately when incident happens.

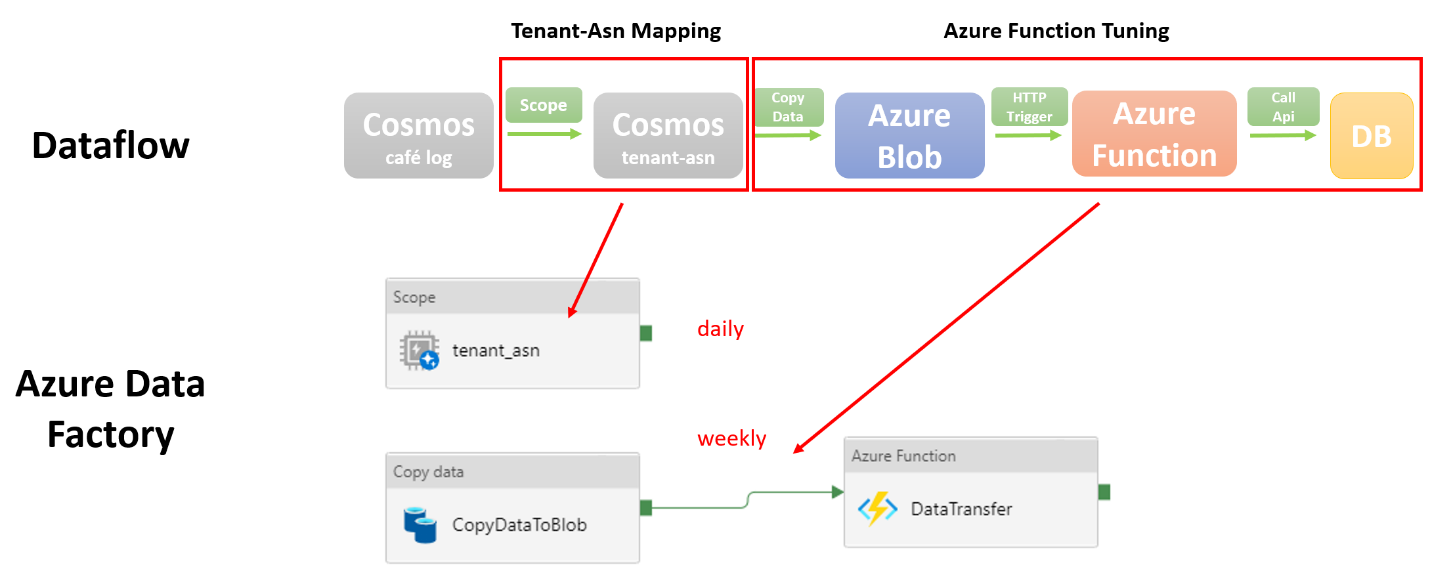
The topology and tenant mapping structure in SHD database will be like this:



The whole pipeline will be divided into two main parts:

1. Scope Script part in Cosmos, it corresponds to the second module below.
2. Azure Function part in Azure Data Factory(ADF), it corresponds to the last three modules below.

The scope will run daily in Cosmos, and the result will be aggregated and push into the DB by Azure Function weekly:



**Scope part: Dataset and Data Flow**

We will use the following 5 main Datasets, and generate the mapping result in Cosmos.

1. **Top Café Log**: the top protocols to generate café log daily, like MapiHttp, Default, Ews, Eas, Rest, Owa, OutlookService, MailboxDelivery.

[COSMOS cosmos/office.adhoc/shares/exchange.storage.prod/local/Resources/Views/Public/Cafe\_SStream.view (osdinfra.net)](https://aad.cosmos14.osdinfra.net/cosmos/office.adhoc/shares/exchange.storage.prod/local/Resources/Views/Public/Cafe_SStream.view?property=info)

2. **IDEAS** **Tenant Dataset** : the tenant related information.

[COSMOS cosmos/office.adhoc/shares/IDEAs.Prod.Data/Publish.Profiles.Tenant.Commercial.IDEAsTenantProfile/Views/v3/IDEAsTenantProfile.view (osdinfra.net)](https://aad.cosmos14.osdinfra.net/cosmos/office.adhoc/shares/IDEAs.Prod.Data/Publish.Profiles.Tenant.Commercial.IDEAsTenantProfile/Views/v3/IDEAsTenantProfile.view?property=info)

3. **Domains**: tenant-domain mapping

[COSMOS cosmos/office.adhoc/shares/IDEAs.Prod.Data/Publish.Profiles.Tenant.Commercial.IDEAsTenantDomainProfile/Views/v1/IDEAsTenantDomainProfile.view (osdinfra.net)](https://aad.cosmos14.osdinfra.net/cosmos/office.adhoc/shares/IDEAs.Prod.Data/Publish.Profiles.Tenant.Commercial.IDEAsTenantDomainProfile/Views/v1/IDEAsTenantDomainProfile.view?property=info)

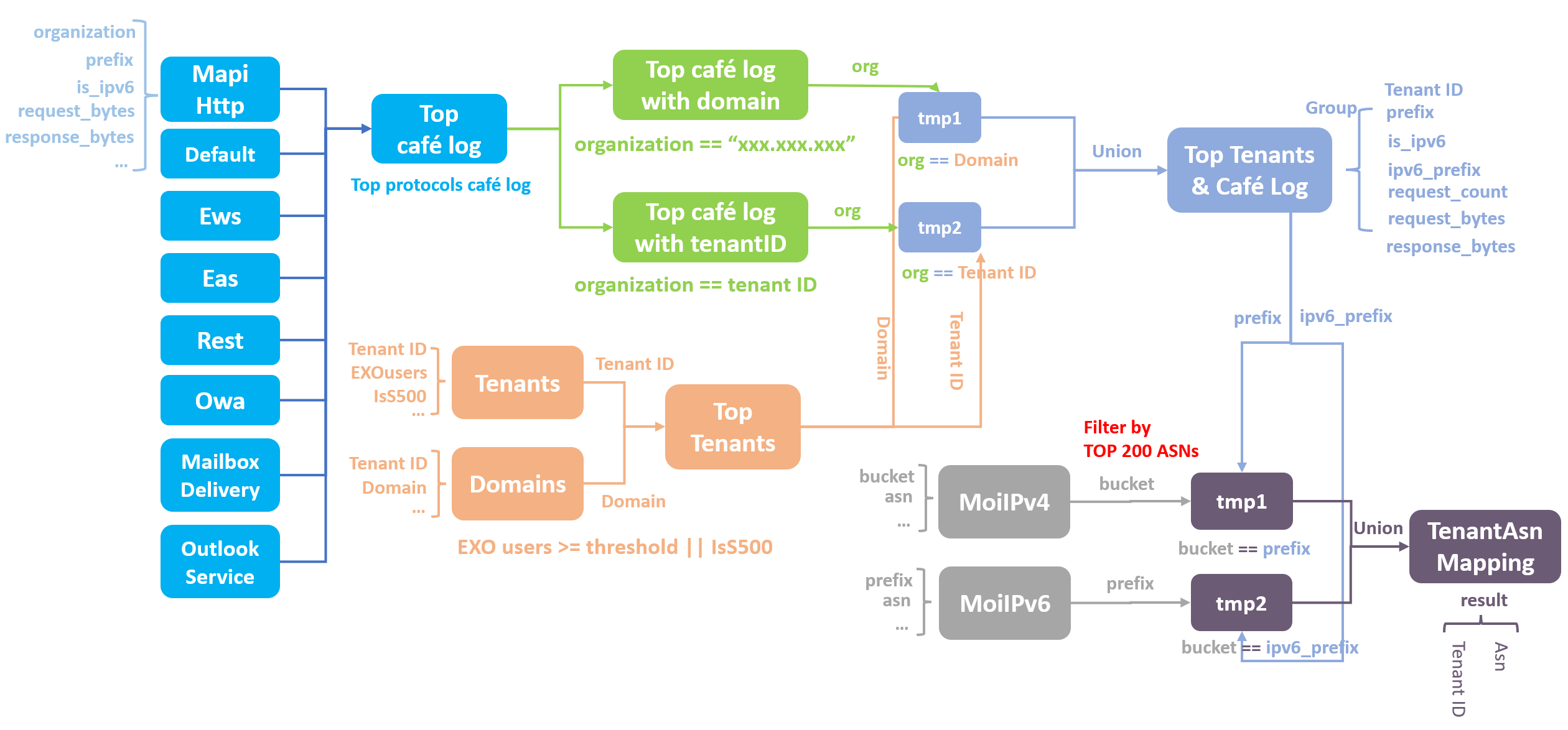
4. **MoiIPv4**: ipv4\_prefix-asn mapping

Eg: [COSMOS cosmos/office.adhoc/shares/Edge.prod/MapOfTheInternet/2021/09/2021-09-01/AugmentedRevIP/AugmentedRevIP.ss (osdinfra.net)](https://aad.cosmos14.osdinfra.net/cosmos/office.adhoc/shares/Edge.prod/MapOfTheInternet/2021/09/2021-09-01/AugmentedRevIP/AugmentedRevIP.ss?property=info)

5. **MoiIPv6**: ipv6\_prefix-asn mapping

Eg: [COSMOS cosmos/office.adhoc/shares/Edge.prod/MapOfTheInternet/2021/09/2021-09-01/AugmentedRevIPv6/AugmentedRevIP.ss (osdinfra.net)](https://aad.cosmos14.osdinfra.net/cosmos/office.adhoc/shares/Edge.prod/MapOfTheInternet/2021/09/2021-09-01/AugmentedRevIPv6/AugmentedRevIP.ss?property=info)

The Scope Script Logic is shown in the figure below: we get the top café log and top tenants, join them to get the tenant-id and prefix relations. And then we get the prefix and asn relations from MoiIPv4/MoiIPv6. Based on the prefix, we can get the tenant-id and asn mapping finally.



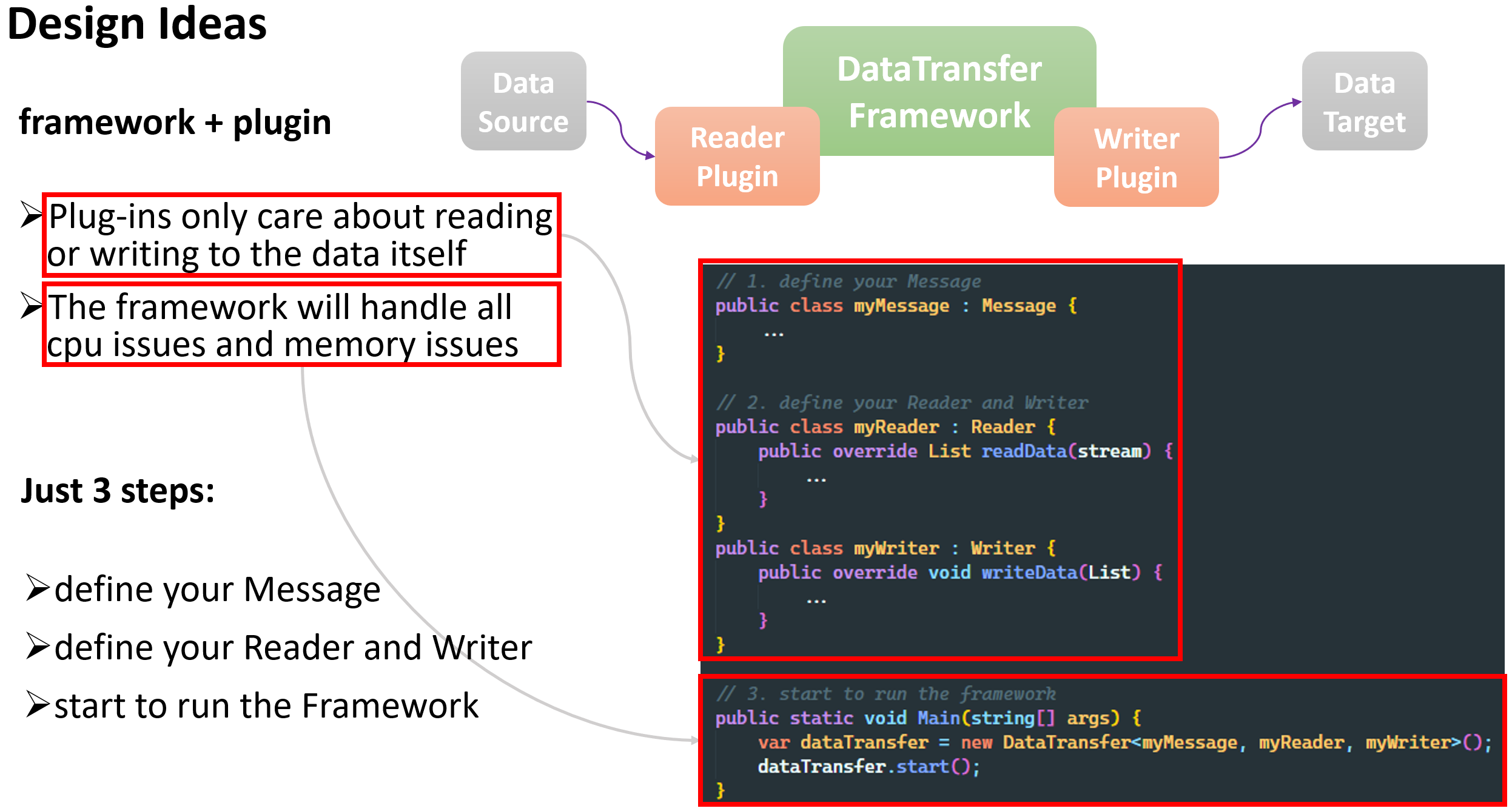
The picture above can help us understand how the scope works, it will generate the Tenant-Asn Mapping in Cosmos, in the form of an intermediate structure stream data in our Office 365 Prod VC: exchange.storage.prod. The scope runs daily, and after a weekly data has been collected, we’ll do the Azure Function part.

**Azure Function part: Data Transfer**

The mapping generated by scope is still kept on cosmos, every week we will push them into the SHD database, just a full refresh update. That’s what Azure Function part need to do.

We’ll use the Azure Data Factory (ADF) to copy the data from Cosmos to Azure Blob Storage firstly. After that, the HTTP trigger will start the Azure Function. The Function will read the data from Azure Blob Storage, and insert these items into the SHD database.

By the way, the Azure Function is designed to be scalable, in a framework + plug-in design pattern. The common part for cpu issues and memory issues is abstracted to be the main code of framework, which benefits for code reuse. It will split the total data into smaller data slices for reading instead of reading all the data at once, to avoid the OOM(out of memory). We also try the different thread models to make the program run more efficiently. For more details, see this link: [Azure Function Tuning](https://microsoftapc-my.sharepoint.com/:p:/g/personal/t-haoge_microsoft_com/EeTtXgypI7pAikit-Ltp3UcBmuYvnaGMYk3AcLI5LQ1wJA?e=MNECUJ).



Finally, we’ll use the Event Authoring API to update data in SHD database. The API link is [Event Authoring Documentation (sharepoint-df.com)](https://microsoft.sharepoint-df.com/teams/M365MCSHDEng/SitePages/Event-Authoring-docs.aspx?OR=Teams-HL&CT=1630905783168#import-topology-tenants). There would be two choices for update: Incremental or Full refresh. We choose the full refresh update policy here, which means all the tenants under the ASN will be replaced and updated once a week, to avoid the case that some old items cannot be deleted successfully under the Incremental policy.

