Organizing subscriptions and resource groups within the Enterprise

Publicado el 26 abril, 2018

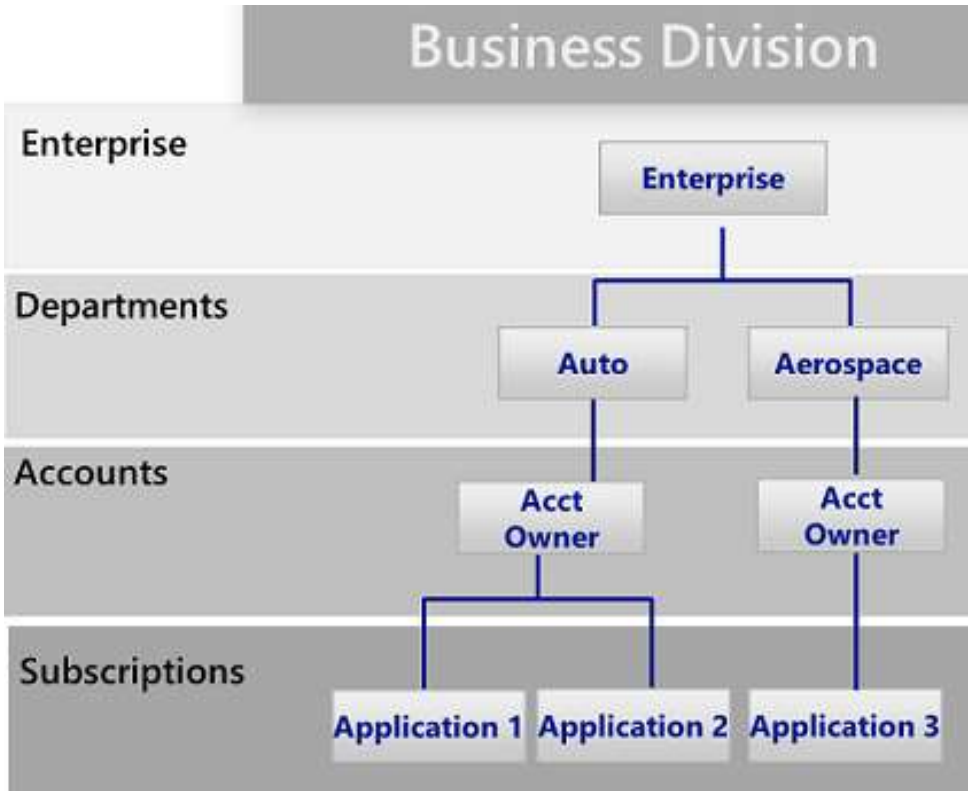
Lyle Dodge
Senior Program Manager, Core Services Engineering

Special thanks to Robert Venable, Principal Software Engineer in the Finance Engineering team of Core Services Engineering (formerly Microsoft IT) for sharing their story of enabling development teams while ensuring security and compliance. Thanks also to Scott Hoag (https://twitter.com/ciphertxt), Principal Cloud Solutions Architect at Opsgility and Rob Dendtler, Account Technology Strategist at Microsoft for reviewing and providing invaluable feedback.

One of the common questions members of the Core Services Engineering and Operations teams frequently get when speaking to customers at the Executive Briefing Center here in Redmond is how do our engineering teams secure our Azure footprint for our Line of Business applications while still giving developers the freedom to go fast, have visibility into our environment and use the capabilities of Visual Studio Team Services for CI/CD, Release, and much more.

At the core of this answer is how we use the combination of subscriptions, resource groups, and Role Based Access Control to ensure compliance with a set of guidelines.

Let's start at the top level: Azure Subscriptions. CSEO, as you can imagine has a lot of Line of Business applications, currently over a thousand. We loosely follow the **business unit** pattern from the Azure enterprise scaffold - prescriptive subscription governance article (https://docs.microsoft.com/azure/azure-resource-manager/resource-manager-subscription-governance/?wt.mc_id=itshowcase-codeapps).



The business unit pattern

In particular, many of our teams have adopted a common mapping of the above pattern to enterprise/federal/state/local. This common vocabulary provides practical constructs that everyone understands and can relate to, ensuring we're on the same page.

What does this translation look like in reality with examples for subscription organization? It looks like this from the top down:

- **Enterprise** - This stays the same as Enterprise in the Azure scaffold for us. Enterprise level items are common concerns across the entire company – it might be ensuring we don't count internal Azure consumption as public revenue, or how secure we are across all Azure subscriptions in our tenants, or other high-level strategic objectives that we care about regardless of level. Another way to think of this might be how Microsoft reports our global quarterly earnings – it's across the entire company.
- **Federal** - Our major departments are named Federal. For example, CSEO is one of the federal groups. At this level, we may have additional policies and procedures, automation that runs against our footprint, or other things specific to that department. Typically, this is where large budgets have roll-up views, etc.

- **State -** A group of services or a service offerings that are related. For example, the Tax Service Offering within the Finance IT organization. Here you may have additional policies and procedures, for example, HIPAA, PCI, SOX controls, and procedures. A state has a group of services that are related.
- **Local** – This is where a subscription lives and is associated with a service. Each subscription contains multiple applications that are related to delivering the set of functionalities that make up the service. Each application is typically contained in an explicit resource group. The resource group becomes the container for that application, which is part of the service (the subscription). There may sometimes be a shared or common application in the service. At the application/resource group level is where the team of application developers live and they're accountable for their footprint in Azure from security to optimal Azure spend in everything they do. A great development team operating at this level solves most of the concerns and roll-up reporting questions that are typically asked from higher levels. If each development team looks at the Azure Security Center blade, pinned dashboards built from Azure Log analytics, and the Azure Advisor blades on a daily basis, you wouldn't have division-wide efforts created to reduce spend or bring up patch compliance, etc.

This hierarchical construct allows the differing level of controls and policies while allowing the developers to go fast. Below is a typical subscription configuration:

(https://azurecomcdn.azureedge.net/mediahandler/acomblog/media/Default/blog/75517cc9-4fe6-4b0a-8976-f0ca058a6548.png)

An example CSEO Subscription for the Tax Service

Within the resource groups above, the typical components you would see in each of the production resource groups (applications) would be the Azure components used to build that specific service such as:

- Azure HDInsight cluster
- Storage Accounts
- SQL Database
- Log Analytics
- Application Insights
- Etc., etc.

Each resource group has the components specific to that application. On occasion, the subscription might have a Common or Shared Services resource group. These are items that are used across the applications in the service, for example:

- Common Log Analytics workspace
- Common Blob Storage accounts where files are dumped for processing by the other services
- An ExpressRoute VNet

In our CSEO Tax Service there are multiple applications, each in their own resource groups such as the data warehouse, ask tax (the help web portal and some bots), a calculation engine, an archiving application, a reporting application and more.

Within the subscription and resource groups, we use least privilege access principles to ensure that only the people that need to do the work have access to resources. Therefore, only the engineering owners of the service are the owners of the subscription. No contributors exist on the subscription. Some specific identities are added to the reader role, these are typically accounts used by automated tooling.

Each resource group has only the identities necessary added with the minimum required permissions. We try to avoid creating custom roles, which over the passage of time and with scale create management headaches.

Within the resource groups, the owner and reader roles are inherited from the subscription. The VSTS Build Account identity is added in as a contributor to the resource group for automated deployments. This means that **only the service owner and the build identities can touch the production service on a continuous basis**.

In the pre-production resource group, the engineering team is added to the reader role. This still means that only the service owner and build accounts can touch pre-production on a continuous basis, but the engineering team can see what's going on in the resource group. If a developer needs to do some work for testing, they can't go putting that in the pre-prod or prod environment.

There are some variations on this but they're not common. For instance, some teams might want someone from the security as the sub owner, and some teams even remove people from the equation and use a form of service account as the sub owner. Some teams might give engineers contributor on pre-prod if they're not fully mature on the automation required. It all depends on the needs of the team.

So now that we have these together, what does it mean for typical roles in the organization?

Developers have access to the pre-production resource group to see what's going on in the dev/pre-production/uat/whatever-you-want-to-call-non-production-in-your-company but must get used to using telemetry mechanisms in the platform for debugging, just like they would have to do in production. While the teams are maturing to this level, you may see developers with contributor level access at the pre-production resource groups. The result of this discipline is typically much richer Application Insights portals and Azure Log Analytics dashboards. As teams mature they switch to

deploying from a CI/CD system like Visual Studio Team Services that uses Microsoft Release Management and get really good at creating build and release definitions. Developers also script out and automate operational concerns like key rotation.

Security & Operations have access via identities with Just-in-Time VM Access through Azure Security Center for IaaS and Privileged Identity Management through Azure AD. Some operations teams might use automated tools to interrogate our Azure subscription footprint looking for configurations to reduce risk, for example looking for a resource group containing a public internet endpoint (PiP) and an ExpressRoute circuit that might pose a security risk. These teams use those audit account identities added at the subscription level.

Another thing that this model implicitly drives is the shift of accountability from any Central IT team to the development team. **This does not mean shift of concerns** as security and operations teams still care about compliance and risk. But if the local development team makes part of their daily standup looking at billing, security center, and the Azure Advisor tools, then cost optimization, security compliance and concerns that are inevitably asked from the enterprise, federal and state layers will already be optimized.

Do you have a question to ask the engineers of Core Services Engineering? You can reach Lyle Dodge on Twitter at @lyledodge (https://twitter.com/lyledodge) and our team will work on the answer to your question in a future article here, or on the IT Showcase site (https://www.microsoft.com/itshowcase).

Developer (/es-es/blog/topics/developer/) Cloud Strategy (/es-es/blog/topics/cloud-strategy/) Subscription Management (/es-es/blog/tag/subscription-management/) Resource Groups (/es-es/blog/tag/resource-groups/) Enterprise (/es-es/blog/tag/enterprise/)

Suscríbase (/es-es/blog/feed/)

Explorar

Vea a dónde nos dirigimos. Compruebe los próximos cambios en los productos de Azure.

Actualizaciones de Azure (/es-es/updates/)

Háganos saber lo que piensa de Azure y lo que le gustaría ver en el futuro.

Proporcionar comentarios (https://feedback.azure.com)

Temas

Announcements (/es-es/blog/topics/announcements/) (2279)

API Management (/es-es/blog/topics/api-management/) (34)

Artificial Intelligence (/es-es/blog/topics/artificial-intelligence/) (231)

Azure Maps (/es-es/blog/topics/azure-maps/) (26)

Azure Marketplace (/es-es/blog/topics/azure-marketplace/) (142)

Azure Stream Analytics (/es-es/blog/topics/azure-stream-analytics/) (31)

Big Data (/es-es/blog/topics/big-data/) (639)

Blockchain (/es-es/blog/topics/blockchain/) (89)

Business Intelligence (/es-es/blog/topics/business-intelligence/) (116)

Cloud Strategy (/es-es/blog/topics/cloud-strategy/) (639)

Cognitive Services (/es-es/blog/topics/cognitive-services/) (126)

Data Science (/es-es/blog/topics/datascience/) (113)

Data Warehouse (/es-es/blog/topics/data-warehouse/) (219)

Database (/es-es/blog/topics/database/) (609)

Developer (/es-es/blog/topics/developer/) (1185)

DevOps (/es-es/blog/topics/devops/) (78)

Events (/es-es/blog/topics/events/) (238)

Government (/es-es/blog/topics/government/) (69)

Hybrid (/es-es/blog/topics/hybrid/) (71)

Identity & Access Management (/es-es/blog/topics/identity-access-management/) (88)

Internet of Things (/es-es/blog/topics/internet-of-things/) (369)

IT Pro (/es-es/blog/topics/it-pro/) (598)

Last week in Azure (/es-es/blog/topics/last-week-in-azure/) (92)

Machine Learning (/es-es/blog/topics/machine-learning/) (41)

Management (/es-es/blog/topics/management/) (353)

Media Services & CDN (/es-es/blog/topics/media-services/) (207)

Migration (/es-es/blog/topics/migration/) (23)

Mobile (/es-es/blog/topics/mobile/) (158)

Monitoring (/es-es/blog/topics/monitor/) (144)

Networking (/es-es/blog/topics/networking/) (220)

Partner (/es-es/blog/topics/partner/) (129)

Security (/es-es/blog/topics/security/) (393)

Serverless (/es-es/blog/topics/serverless/) (73)

Storage, Backup & Recovery (/es-es/blog/topics/storage-backup-and-recovery/) (682)

Supportability (/es-es/blog/topics/supportability/) (46)

Updates (/es-es/blog/topics/updates/) (571)

Virtual Machines (/es-es/blog/topics/virtual-machines/) (700)

Web (/es-es/blog/topics/web/) (364)

Artículos por fecha

febrero 2021 (/es-es/blog/2021/02/)

enero 2021 (/es-es/blog/2021/01/)

diciembre 2020 (/es-es/blog/2020/12/)

noviembre 2020 (/es-es/blog/2020/11/)

octubre 2020 (/es-es/blog/2020/10/)

septiembre 2020 (/es-es/blog/2020/09/)

Archivo completo (/es-es/blog/archives/)