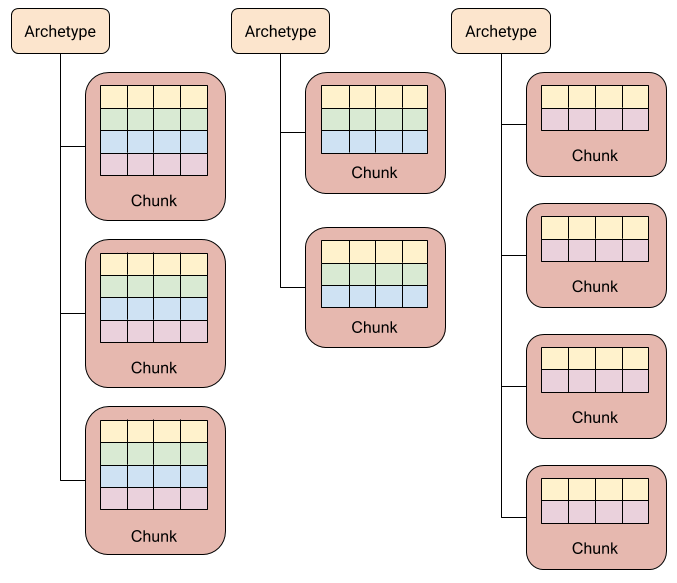
**Components**

Components represent the data of your game or application. [Entities](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/ecs_entities.html) are identifiers that index your collections of components, while [systems](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/ecs_systems.html) provide the behavior.

A component in ECS is a struct that has one of the following "marker interfaces":

* [IComponentData](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.IComponentData.html) — Use for [general purpose](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/component_data.html) and [chunk components](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/ecs_chunk_component.html).
* [IBufferElementData](xref:Unity.Entities.IBufferelementData) — Associates [dynamic buffers](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/dynamic_buffers.html) with an entity.
* [ISharedComponentData](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.ISharedComponentData.html) — Categorizes or groups entities by value within an archetype. For more information, see [Shared Component Data](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/shared_component_data.html).
* [ISystemStateComponentData](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.ISystemStateComponentData.html) — Associates a system-specific state with an entity and detects when individual entities are created or destroyed. For more information, see [System State Components](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/system_state_components.html).
* [ISharedSystemStateComponentData](xref:Unity.Entities.ISharedSystemStateComponentData) — a combination of shared and system state data. See [System State Components](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/system_state_components.html).
* [Blob assets](xref:ecs-blob-asset-data) – While not technically a "component," you can use blob assets to store data. Blob assets can be referenced by one or more components using a [BlobAssetReference](xref:Unity.Entities.BlobAssetReference) and are immutable. You can use blob assets to share data between assets and access that data in C# jobs.

The EntityManager organizes unique combinations of components into **archetypes**. It stores the components of all entities with the same archetype together in blocks of memory called **chunks**. The entities in a given chunk all have the same component archetype.



Shared components and chunk components are exceptions because ECS stores them outside of the chunk. A single instance of these component types apply to all of the entities in the applicable chunks. Additionally, you can optionally store dynamic buffers outside of the chunk. Even though ECS does not store these types of components inside of the chunk, you can generally treat them the same as other component types when querying for entities.