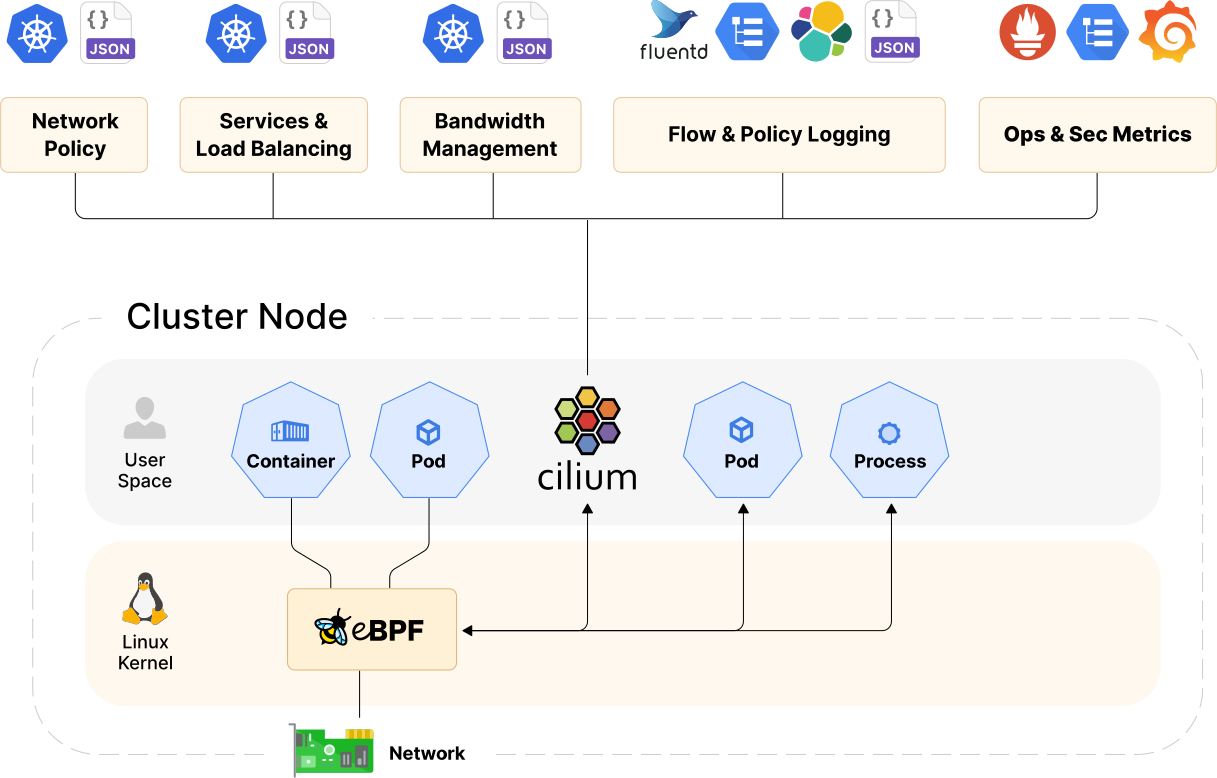
**Cilium Stack**

Components:

1. **Cilium**: It’s the core of the stack
2. **eBPF**: For the dynamic insertion of node.
3. **BPF** **Maps**: A DS in eBPF to communicate b/w user and eBPF programs
4. **Cilium** **CLI**: To interact and manage cilium deployements.
5. **Cilium Agent**: runs on each node in the K8s cluster and is responsible for installing and managing the eBPF programs required for Cilium's networking and security functionalities.
6. **Cilium Proxy**: an optional component that integrates with popular service mesh solutions like Istio and Envoy



**Why Cilium?**

There are lots of CNI options available today. We wanted to stick to eBPF, which proved to be a powerful technology providing many benefits in terms of observability, security, etc. With that in mind, two well-known projects arise when you think of CNI plugins: Cilium and Calico.

1. **Performance**: Using bpfilter (instead of iptables) for routing means shifting filtering tasks to the kernel space, which yields impressive gains in performance
2. **Better network policies**
3. **Inter-node traffic control**: using CiliumClusterwideNetworkPolicy we can control the policies for inter node in the complete cluster
4. **Policy enforcement modes**: It’s easy-to-use policy enforcement modes. The default mode is a good fit in most cases: with no initial restrictions, but as soon as something is allowed, all the rest becomes restricted. Always mode — when policy enforcement is on for all endpoints — is helpful for environments with higher security requirements.
5. **Visual Policy Editor**: provides an easy-to-use, mouse-friendly UI to create the rules and get the corresponding YAML configuration to apply them

