EVE Labs

All ongoing discussions and gory details for the above are kept in EVE's Pivotal Tracker

- Jetson GPU virtualization (**Dmitrii Aleksandrov**)
  - Document architecture and design for EVE support of the Jetson GPU.
  - Deliver working prototype for a Jetson GPU integrated with an application running on EVE.
- Initial NXP bringup (**Dmitrii Aleksandrov**)
  - Investigate EVE running on a small footprint for installation in vehicles.
  - Deliver whitepaper on technology and architecture for “EVE in a car”
- Modularize EVE (**Petr**)
  - Remove the complexity associated with using pillar.
  - Restructure all of the commands under pkg/pillar/cmd/ such that they can run as completely standalone containers.
- Non TPM based security solution (**Avi**)
  - Research and deliver proposal on alternative security approaches for EVE deployments where TPM is not supported
  - Implement PoC of alternative security architecture.
- EVE Kubernetes integration (**Avi**)
  - What does it mean to integrate EVE with Kubernetes, not running Kubernetes *on top of* EVE (k3s solutions already exist for this), but rather how do we integrate EVE the platform with Kubernetes? There are several ways to look at this. Each of these needs to be explored for feasibility and usefulness. It also needs to ensure that it actually covers all of the EVE use cases.
    1. EVE as kubelet: Can a Kubernetes cluster communicate directly with an EVE device? In what scenarios does that make sense? What would that flow look like? The user-controller interaction is at the Kubernetes API level (e.g. kubectl and manifests), while the EVE device looks like a normal node running kubelet.
    2. EVE controller (Adam) exposing a Kubernetes API, i.e. looking and feeling like a kube-apiserver. The controller-to-EVE API remains the current EVE-specific API.
    3. EVE controller embedded into a Kubernetes API. The controller interface is an actual kubernetes cluster, i.e. kube-apiserver. EVE controller is deployed as an application running on top of Kubernetes. This is distinct from the current standalone single process (e.g. OSS Adam) or cloud service (e.g. commercial Zedcloud). It is just a series of OSS components that are deployed to Kubernetes. User-controller is via (possibly extended) Kubernetes API commands; controller-EVE is via current EVE-specific API.
    4. More?