ONOS Software Modularity

**Modularity**
- Increases architectural coherence, testability, maintainability
- Facilitates extensibility & customization by partners & users
- Avoids speciation of the ONOS codebase
- Preempts code entanglement, i.e. cyclic dependencies

**Protocol Independence**
- Raises level of abstraction
- Protects apps from lower level (tables, flow conflicts)
- Allows support for different versions of OpenFlow
- Facilitates gradual migration to SDN

### System Tiers

- **Applications**
- **Northbound API**
- **Core**
  - Device, Host, Link, Topology, Path, Flow, Network...
- **Southbound API**
- **Adapters**
  - Device, Host, Link, Flow, Packet
- **Protocols**
- **Network Elements**

### System Modules

- **Well-defined relationships between system modules**
- **Basis for customization and extensibility**
- **Mechanism to avoid cyclic dependencies**

### Subsystem Structure

- **App Component**
  - command
  - query & command
- **Listener**
  - notify
  - add & remove
- **Store**
  - sync & persist
- **AdminService**
  - Manager Component
  - command
  - sensing
- **Service**
  - ProviderService
  - ProviderRegistry
  - register & unregister
- **Provider**
  - Protocols
- **Provider Component**

ONOS Partners: