ϕ — Single Guard, No Policies

- NSA and BAH models here
  - NSA two domains with filter chains
  - BAH filter segment evaluates data between interface segments
- Usually one domain more sensitive than the other
  - need-to-know
  - not always, sometimes compartmented data just can't be mixed
- rules tightly coupled to filter implementations, filters need to know *specific information* about content

α — Single Guard, Policy Integration

- Ratheon and NSA SOA models here
  - Both use explicit policies to guide decisions
- at least two domains, could have more
- separating obligations, permissions, constraints gives flexibility
- filters can be more generic, supporting policies defined with a specific ontology
- more flexible
  - rules now described in policies that accompany content
  - guards no longer need to know how specific content can be used *a priori*
  - just need to understand a fairly static metamodel, then get specifics from policies
  - e.g. here guard knows that content may have expiration dates; in ϕ guard needs to know the expiration data *a priori*
- easier to manage, maintain
- policies can be referenced via URI or packaged with content directly
- first external service, others can exist

β — Router Guards

- Using overlays, but could be implemented in physical elements or SDN-like systems
- Each domain hosts network, could be degenerate of one node
- Content evaluation at routing points
- Some traffic easier than others
  - HTTP easier than TCP
- More security in depth
- Can create compartments under specific routers (ad-hoc creation!); beginnings of multi-tenancy

γ — Router and Node Guards

- Integrated with routers and nodes
- evaluate at all levels; emitting requests, fielding requests, and in between
- Single node compartments possible
- more security in depth