

# YAHOO!

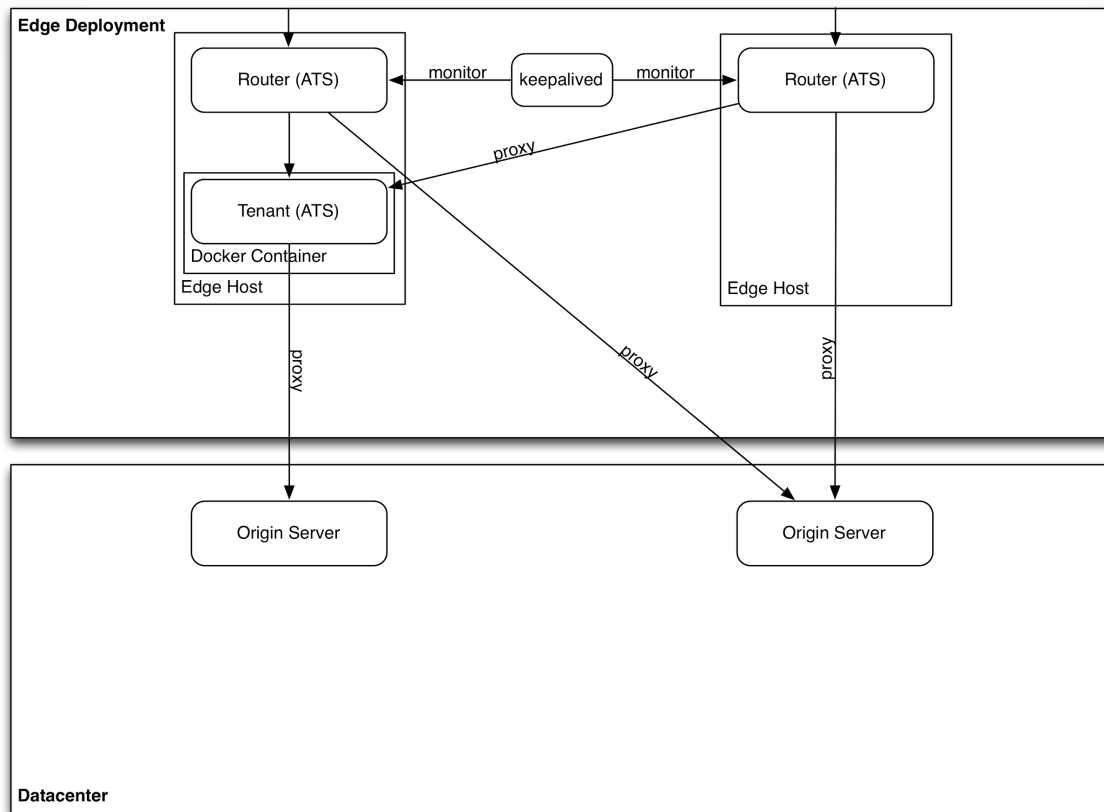
Edge PaaS 2015

Joshua Blatt

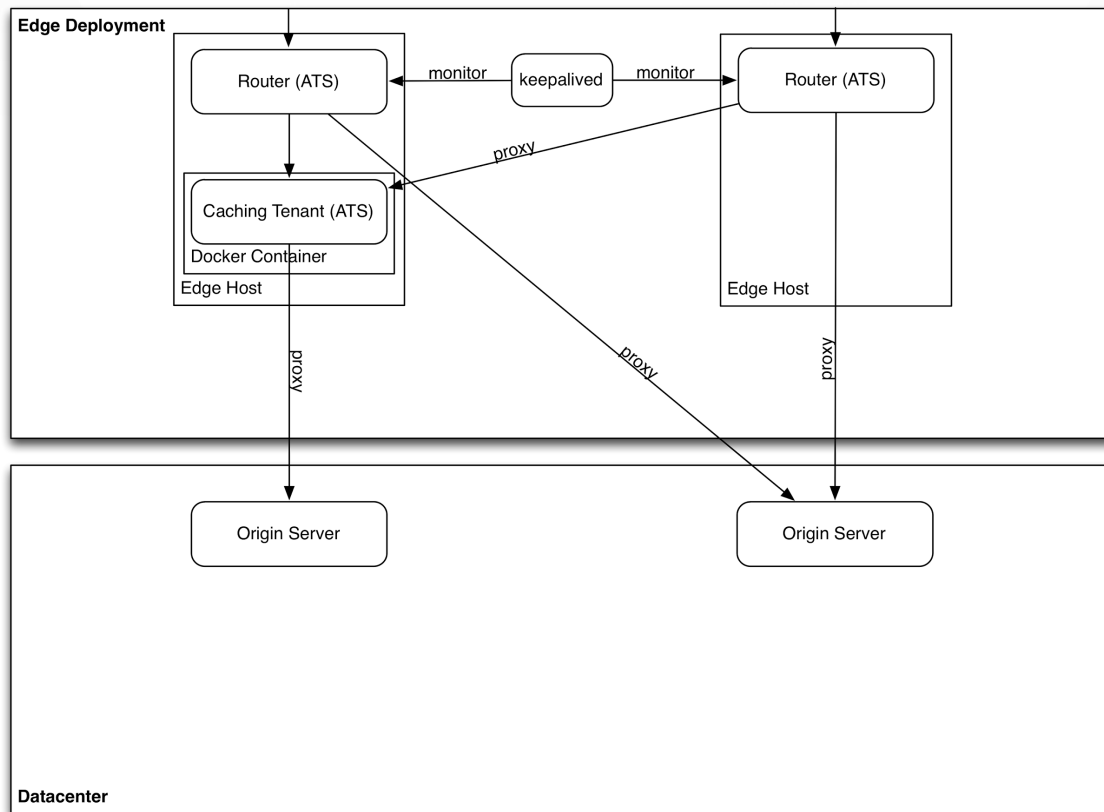
# Yahoo! Challenges

- No Isolation Between Tenants
- No Self Service
- Resulting Service Fragmentation / Security

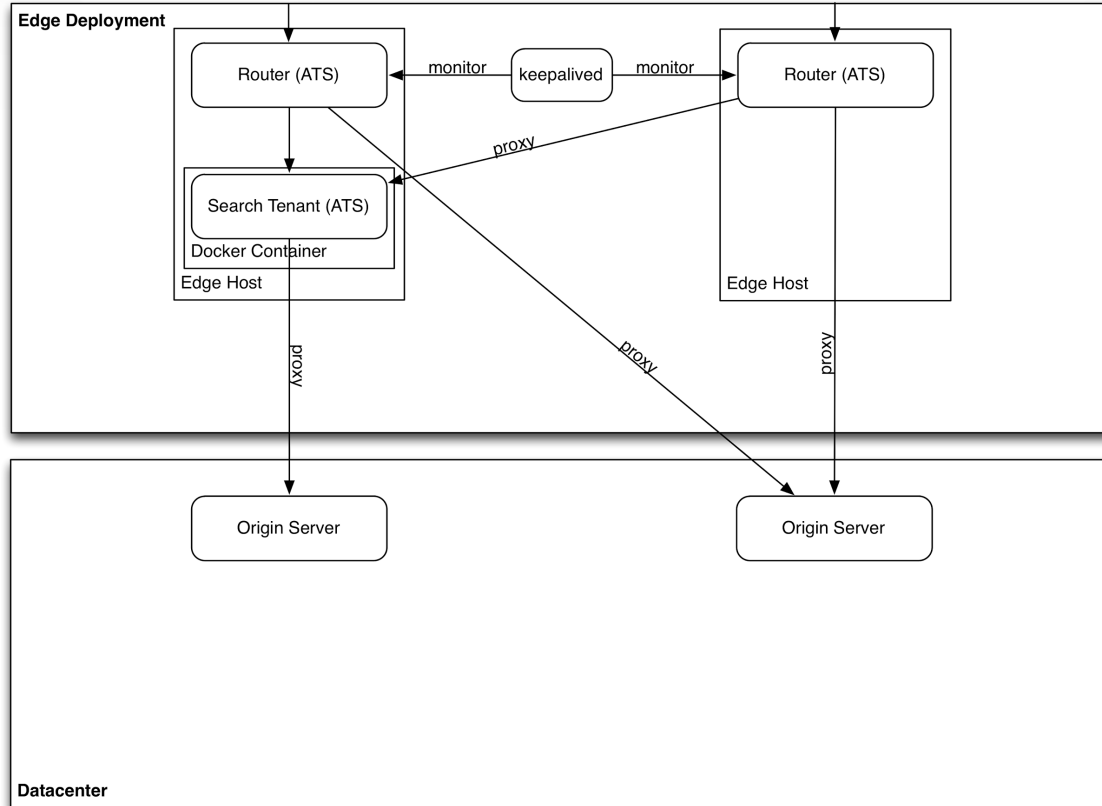
# Edge Data Plane



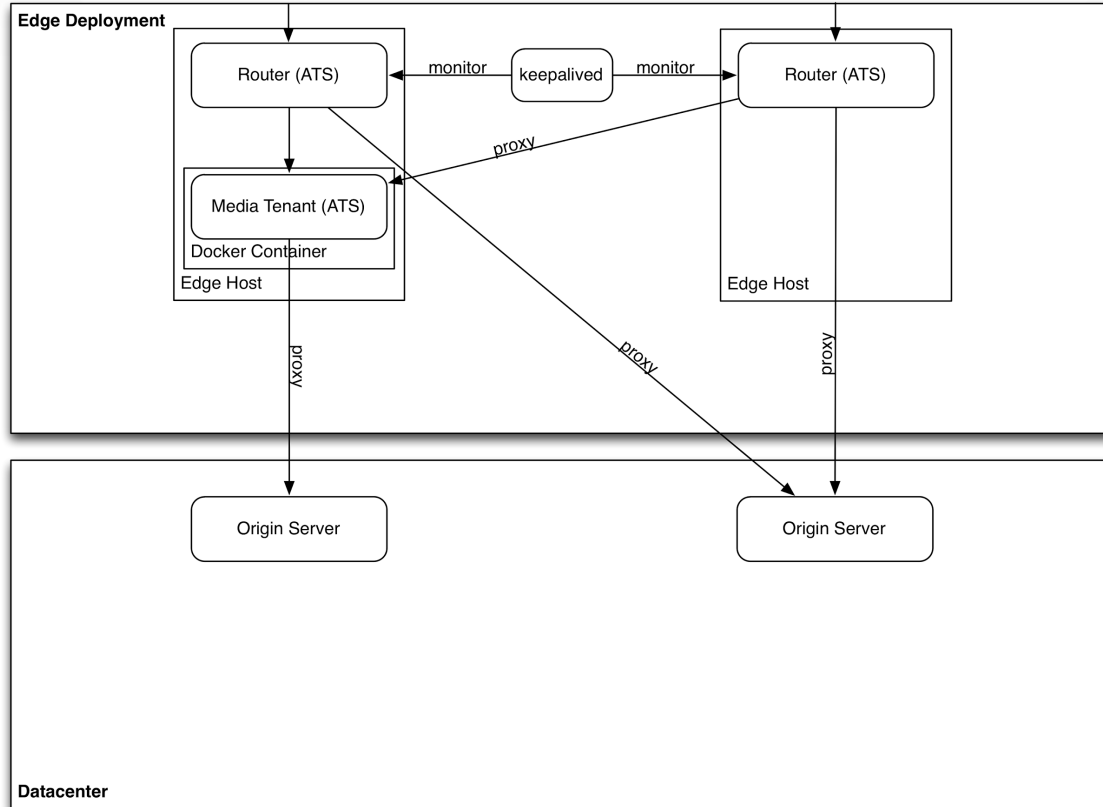
# Microkernel: Caching Tenant



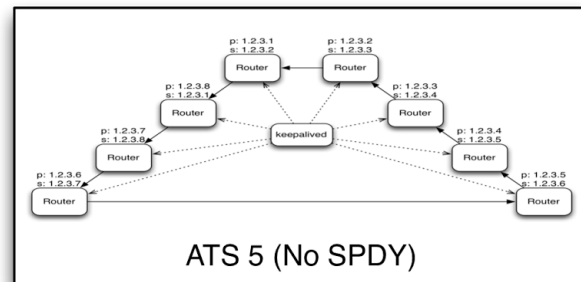
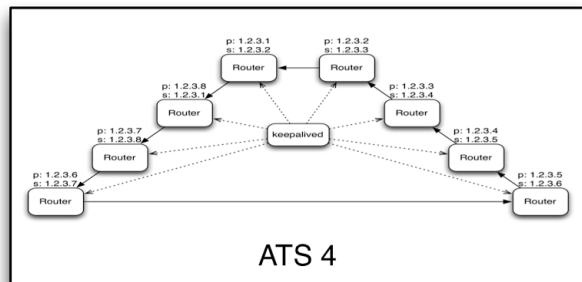
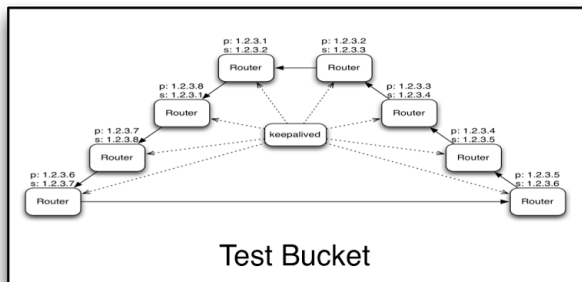
# Application Tenants



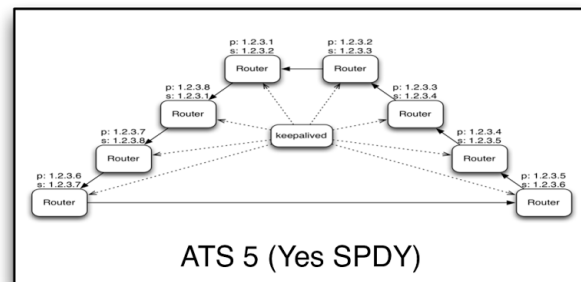
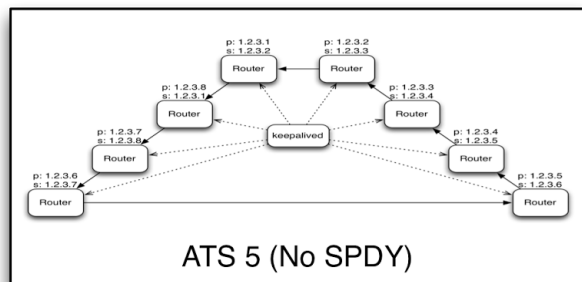
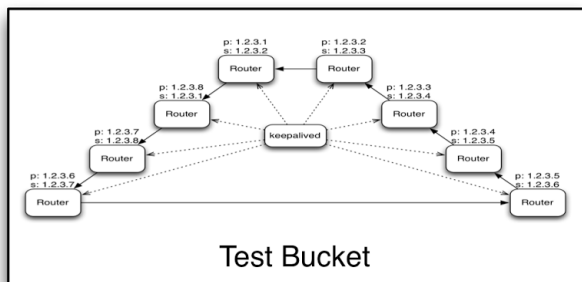
# Application Tenants



# Multiple Routing Tiers Per Edge Deployment (Ideal Past)

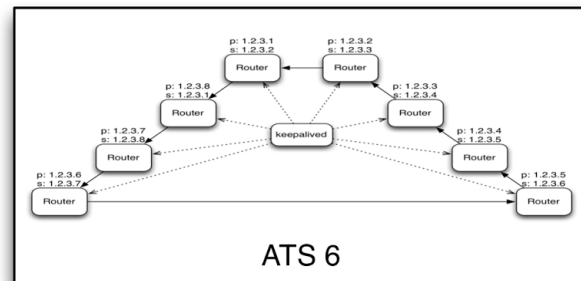
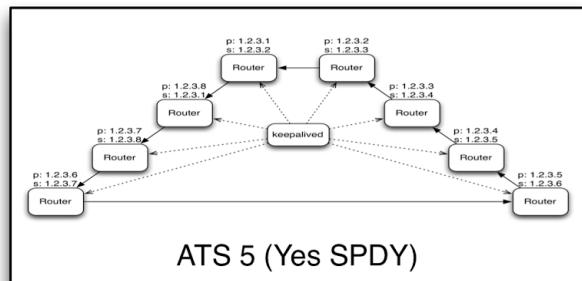
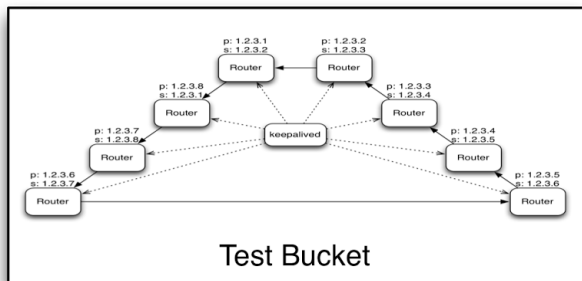


# Multiple Routing Tiers Per Edge Deployment (Ideal Present)

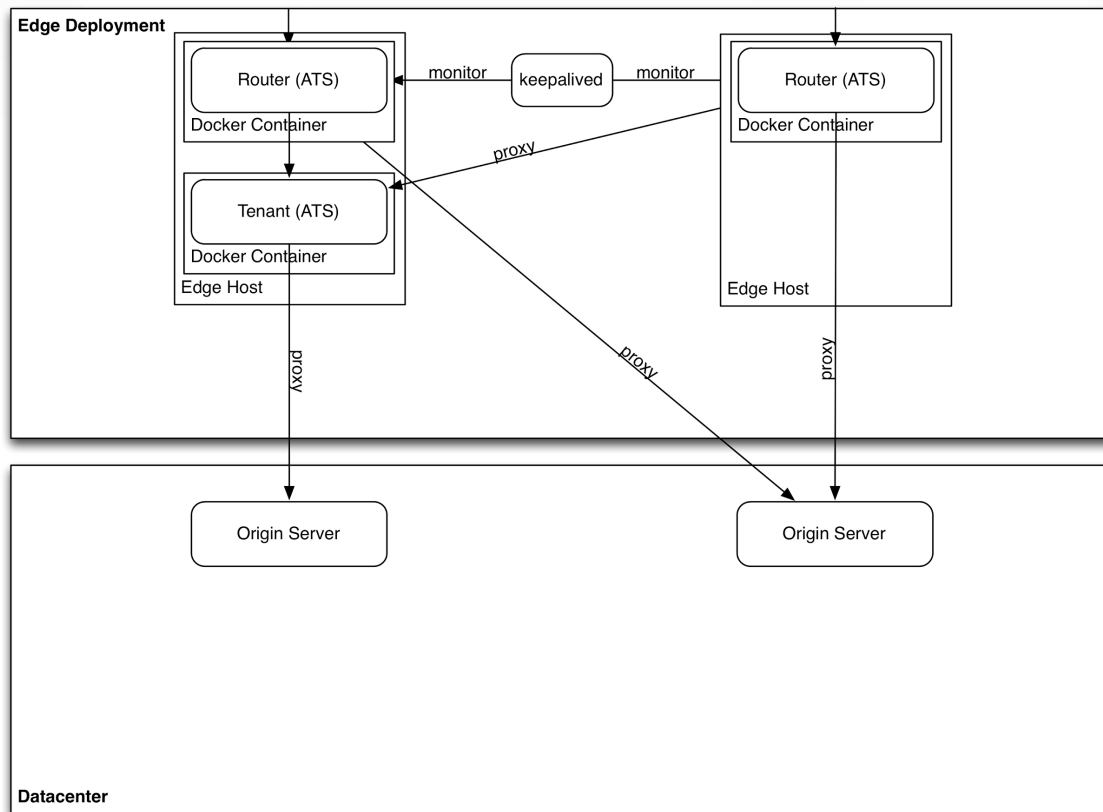




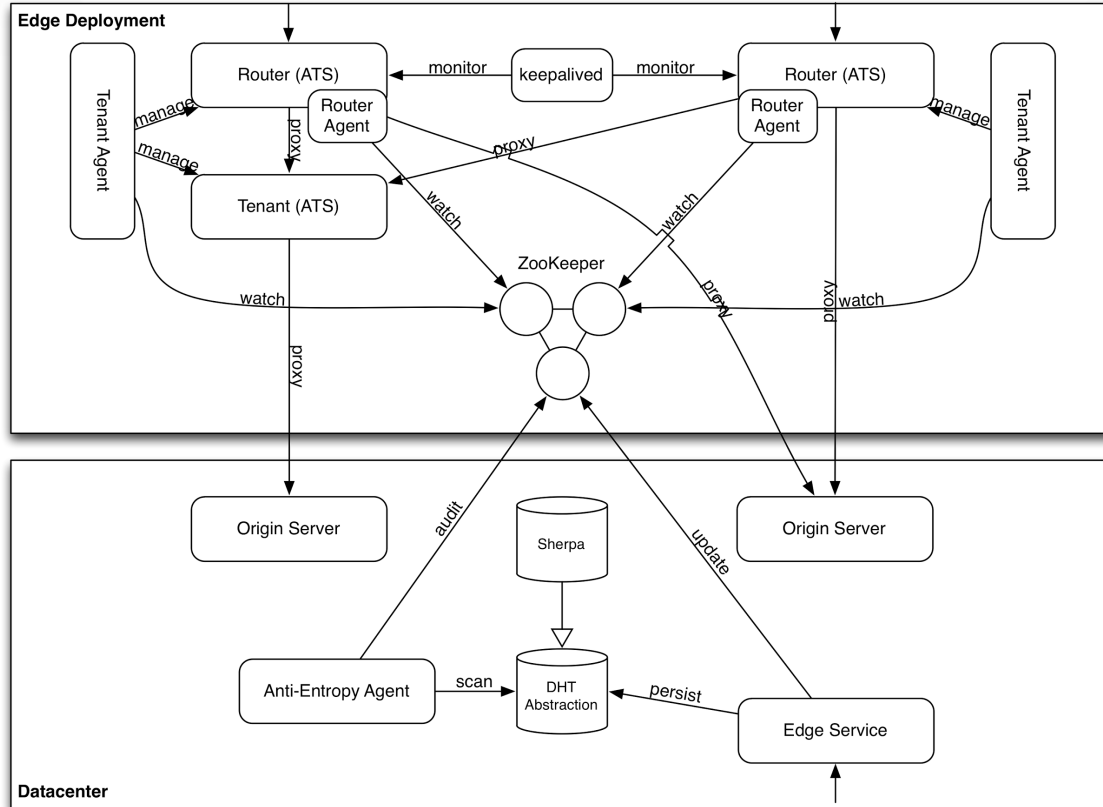
# Multiple Routing Tiers Per Edge Deployment (Future)



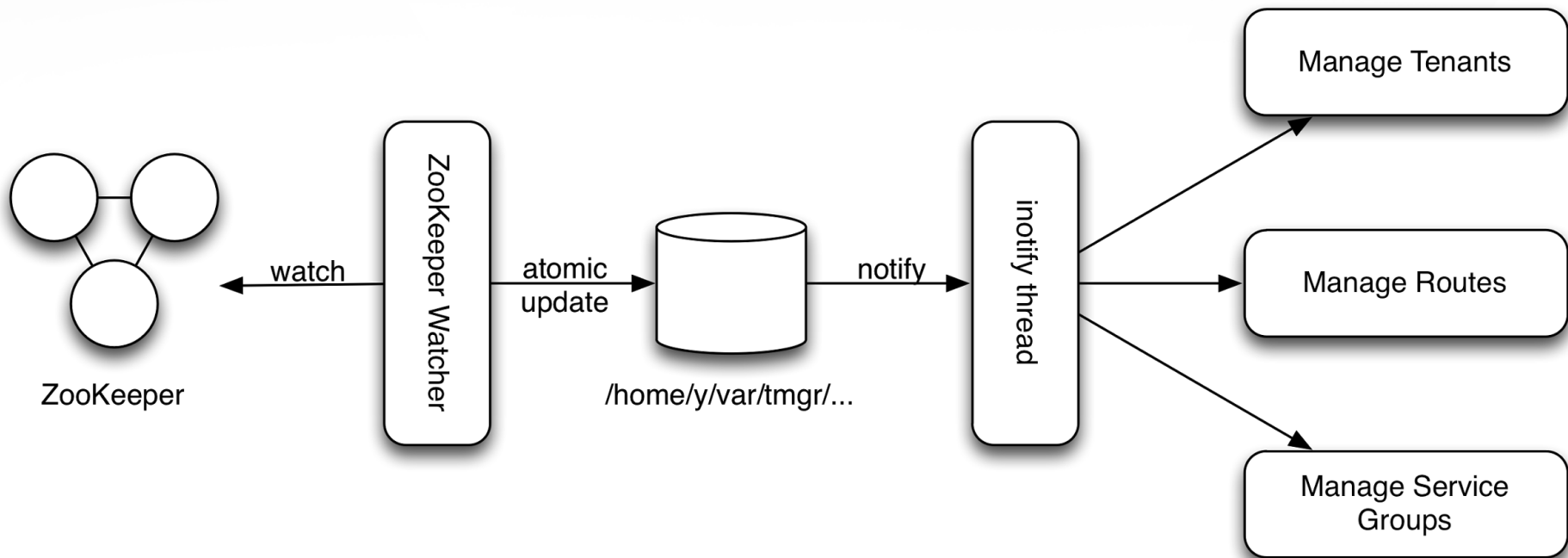
# Microkernel: Routing Tier as a Tenant?



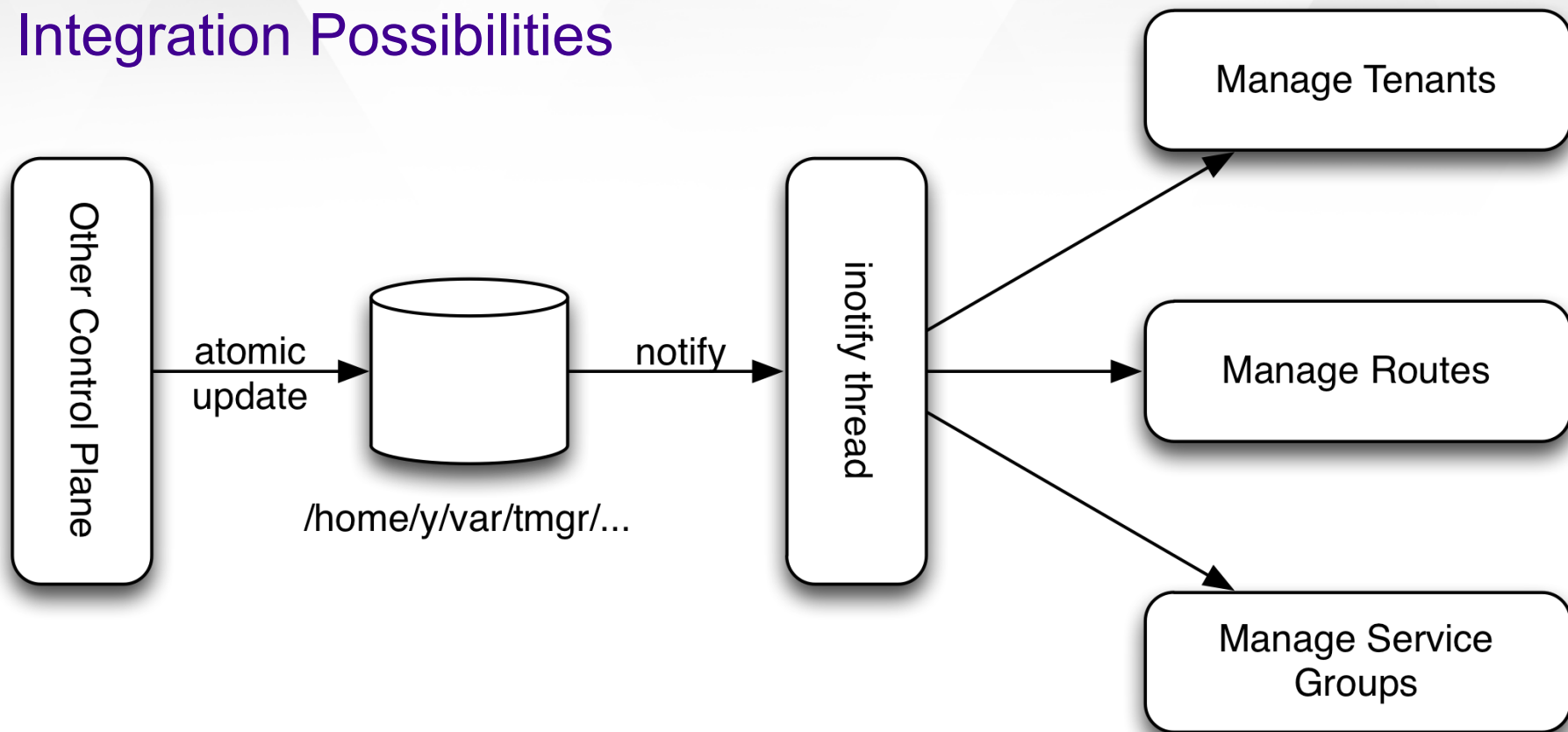
# Edge Control Plane



# Implementation Pattern for Tenants, Routes, Services



# Integration Possibilities



# Tenants

- ATS Base Image
- Tenant:
  - ATS Plugins
  - ATS Config Files (environment specific)

# Tenant Management

- Create Container
- Install ATS (Base) + ATS Plugins (Tenant)
- Start Container
- Stop Container
- Upgrade ATS (Base) + ATS Plugins (Tenant)
- Start Container
- Stop Container
- Destroy Container

# Route Management: Routing Language

- **Directive:** a rule for recognizing and routing a subset of traffic appropriately
- **Filter:** a component part of a directive that matches or samples part or all of a domain's transaction stream.
- **Modifier:** a part of a directive that specifies an action, such as insertion of a header field or addition of a tag, to be performed on a selected request. Essentially, a filter with side-effects.
- **Target:** a part of a directive that specifies a server, server group, or tee to receive traffic for a selected request.

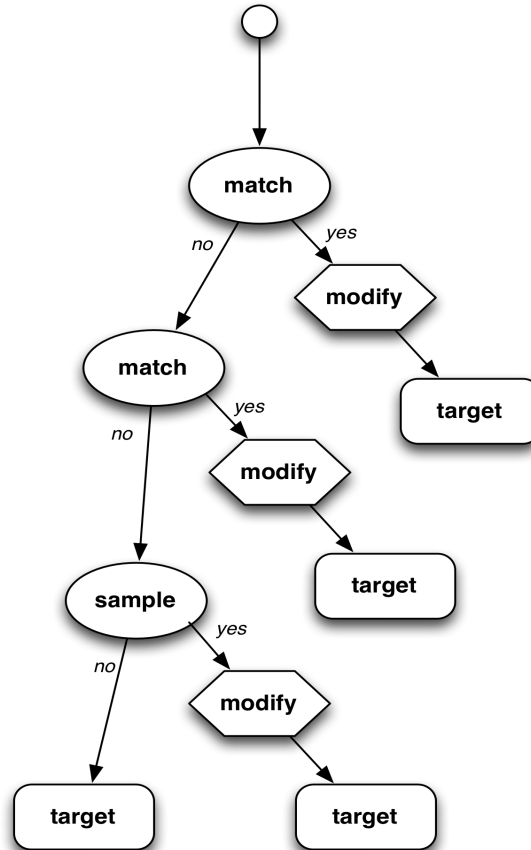


# Directives

```
{ "selector" : {                                # tee or route
  "param-name" : "argument",                    # zero or more
  "filters" : [ {                                # filter list (optional)
    "filter-name" :                             # mandatory per filter
    { "filter-param" : "argument" } # zero or more
  } ],
  "modifiers" : [ {                              # modifier list (optional)
    "modifier-name" :                            # mandatory per filter
    { "modifier-param" : "argument" } # zero or more
  } ],
  "target" : "destination" # mandatory if incomplete filtering
  "tee" : "destination"    # optional; dups for observation
}
}
```

# Example

```
{ "filters" : [  
  { "match" :  
    { "prefix" : "/one",  
      "modifiers" : [  
        { "insert" : {  
          { "header" : "X-mark",  
            "value" : "group1"  
          }  
        }  
      ]  
    },  
    "target" : "grp1"  
  },  
  { "match" :  
    { "prefix" : "/two",  
      "modifiers" : [  
        { "insert" : {  
          { "header" : "X-mark",  
            "value" : "group2"  
          }  
        }  
      ]  
    },  
    "target" : "grp2"  
  },  
  { "sample" :  
    { "fraction" : 0.05,  
      "source" : "bcookie"  
      "modifiers" : [  
        { "insert" : {  
          { "header" : "X-bucket",  
            "value" : "test"  
          }  
        }  
      ]  
    },  
    "target" : "test-bucket"  
  },  
  "target" : "the-rest"  
}]  
}
```



# Filters

```
{ "selector" : {                                # tee or route
  "param-name" : "argument",                  # zero or more
  "filters" : [ {                              # filter list (optional)
    "filter-name" :                          # mandatory per filter
    { "filter-param" : "argument" } # zero or more
  } ],
  "modifiers" : [ {                            # modifier list (optional)
    "modifier-name" :                        # mandatory per filter
    { "modifier-param" : "argument" } # zero or more
  } ],
  "target" : "destination" # mandatory if incomplete filtering
  "tee" : "destination" # optional; dups for observation
}
}
```

# Filters: Match Filters

```
{ "route" :  
  { "filters" : [  
    { "match" : {  
      "type" : "url",  
      "prefix" : "/interesting",  
      "target" : "interesting-group"  
    }  
  ]  
}
```

## Filters: Match Types

- **url**
- **header**
- **bcookie**
- **ycookie**
- **woeid**
- **sid**
- **tag**

## Filters: Match Strings

- **prefix**
- **number** (compared numerically)
- **pattern**

## Filters: Sampling

```
{ "sample" :  
  { "fraction" : 0.0-to-1.0,  
    "partition" : 1-or-greater,  
    "source" : <"bcookie"|"random"|"weoid"|"sid" |  
              "bucket", "tag", "bcookie">,  
    "target" : "destination"          # optional  
  }  
}
```

## Filters: Sample Example

```
{ "route" :  
  { "filters" : [  
    { "sample" :  
      { "fraction" : 0.05,  
        "source" : "bcookie"  
      }  
    } ],  
    "target" : "fivepercent"  
  },  
  "target" : "therest"  
}
```



# Modifiers

```
{ "selector" : {                                # tee or route
  "param-name" : "argument",                    # zero or more
  "filters" : [ {                                # filter list (optional)
    "filter-name" :                             # mandatory per filter
    { "filter-param" : "argument" } # zero or more
  } ],
  "modifiers" : [ {                              # modifier list (optional)
    "modifier-name" :                            # mandatory per filter
    { "modifier-param" : "argument" } # zero or more
  } ],
  "target" : "destination" # mandatory if incomplete filtering
  "tee" : "destination"    # optional; dups for observation
}
}
```

## Modifiers: Insert

```
{ "insert" : {  
  "header" : "X-marker",  
  "value" : "$marker",  
  "following" : "Date"  
}  
}
```

## Modifiers: Delete

```
{ "delete" : {  
  "header" : "X-marker"  
}  
}
```

## Modifiers: Replace

```
{ "modify" : {  
  "header" : "X-marker",  
  "pattern" : "foo*bar",  
  "replacement" : "whatisit"  
}  
}
```

## Modifiers: Append

```
{ "modify" : {  
  "header" : "Set-Cookie",  
  " append" : "; HttpOnly"  
}  
}
```

# Targets

```
{ "selector" : { # tee or route
  "param-name" : "argument", # zero or more
  "filters" : [ { # filter list (optional)
    "filter-name" : # mandatory per filter
    { "filter-param" : "argument" } # zero or more
  } ],
  "modifiers" : [ { # modifier list (optional)
    "modifier-name" : # mandatory per filter
    { "modifier-param" : "argument" } # zero or more
  } ],
  "target" : "destination" # mandatory if incomplete filtering
  "tee" : "destination" # optional; dups for observation
}
}
```

## Targets: Server Groups

- 1+ hosts, containers, virtual machines all running same software
- aka Server Groups, Host Groups, Clusters, Tiers, Roles
- Membership is highly dynamic
  - Active Monitoring: Polling
  - Passive Monitoring: Proxying Fails
  - Sensitive to network partitions, oscillations.

# Targets: Server Groups 1

```
{ "name" : "server-group-name",
  "precedence" : [ {
    "name" : "member-name",           # (opt) name of group member
    "qps-total" : number,             # (opt) max qps across servers
    "qps-server" : number,           # (opt) max qps for a server
    "min-server" : number,           # (opt) minimum active servers
    "fail-rate" : number,            # (opt) fraction fail allowed
    "concurrency" : number,          # (opt) inflight req./server
    "timeout" : number,              # (opt) max time 1st byte ms
    "completion-time" : number,      # (opt) max time req complete
    "outstanding" : number,          # (opt) max req outstanding
    "startup-rate" : number,         # (opt) slow-start ramp time ms
    "recover-rate" : number,        # (opt) time back to full bw
    "overload" : "failover"
      | " spillover"
      | " drop"           # (opt) ovrlb behavior
```

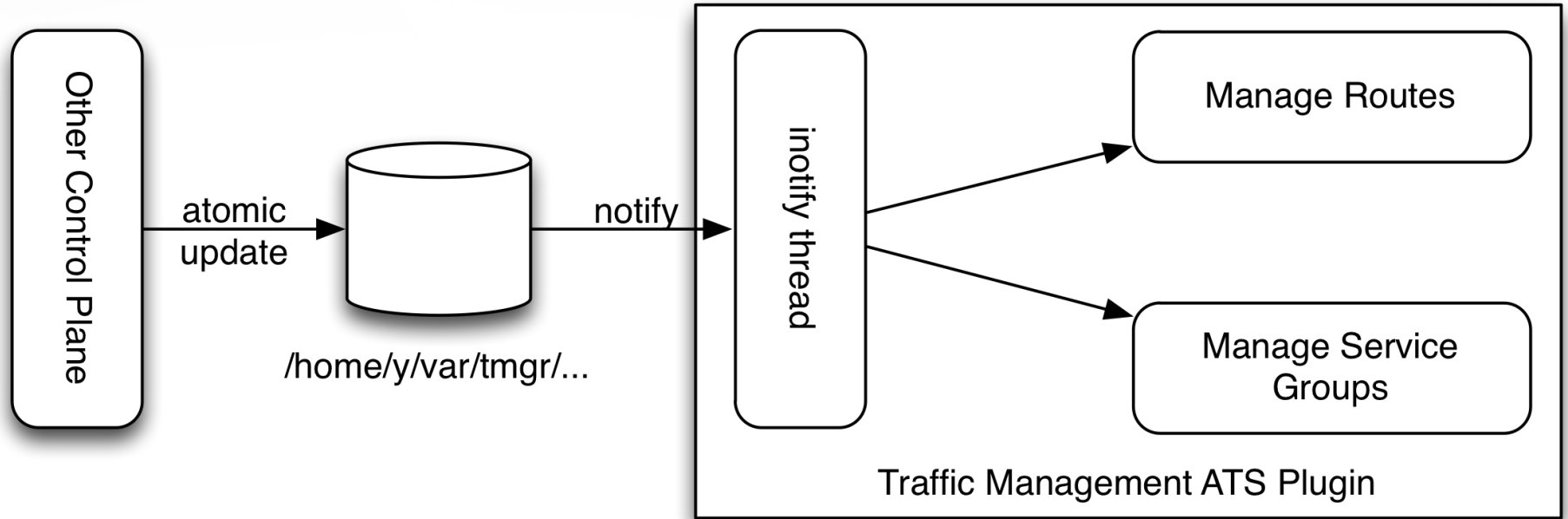


# Targets: Server Groups 2

```
"routing" : "round-robin"      # (sequential)
          | "balanced"        # (least-busy)
          | "random"          # (opt) routing/rotation scheme
"servers" : [
  { "name" : destination, "port" : number,
    "protocol" : protocol, "weight" : number },
    # port & protocol default to
    # passthru; weight is optional

  { "name" : destination, "port" : number,
    "protocol" : protocol, "weight" : number },
    "alternates" : [ "service", "service" ],
    # alternate service groups
  | "redirect" : "redirect-url" # where to go on fail/overload
  | "internal" : "local-url"   # internal (reprocess)
  | "builtin" : "path"
},
{ . . . } ]
}
```

# Integration Possibilities



# Edge Control Plane

