Self-Service Architecture

October 18, 2017

Goals

1. Privileged users should be able to add/modify/delete:
    A. Other users in their tenant
    B. Sub-tenants beneath their tenant
    C. Delivery services in their tenant

Goals

1. Privileged users should be able to add/modify/delete:

   A. Other users in their tenant

   B. Sub-tenants beneath their tenant

   C. Delivery services in their tenant

Goals

1. Privileged users should be able to add/modify/delete:

   A. Other users in their tenant

   B. Sub-tenants beneath their tenant

   C. Delivery services in their tenant

Goals

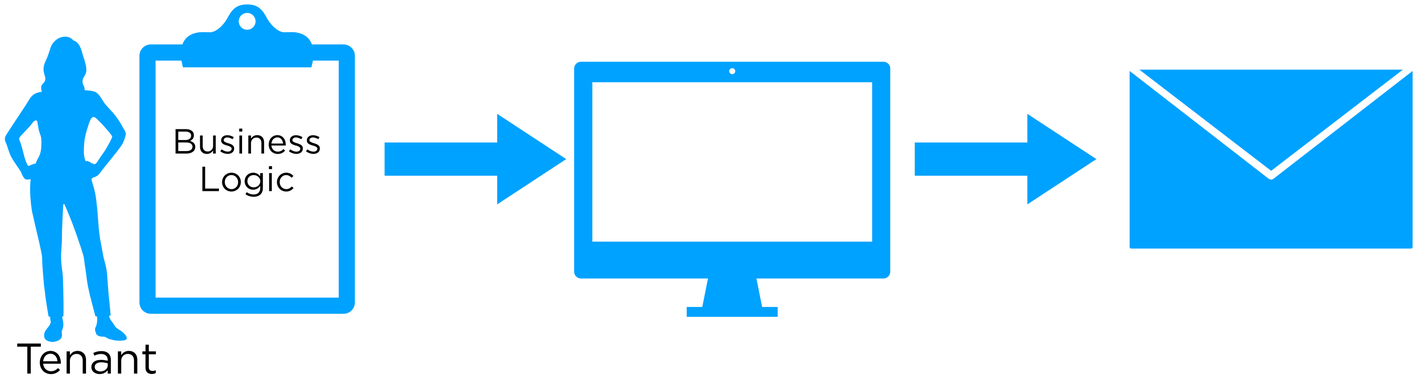
1. Privileged users should be able to add/modify/delete:

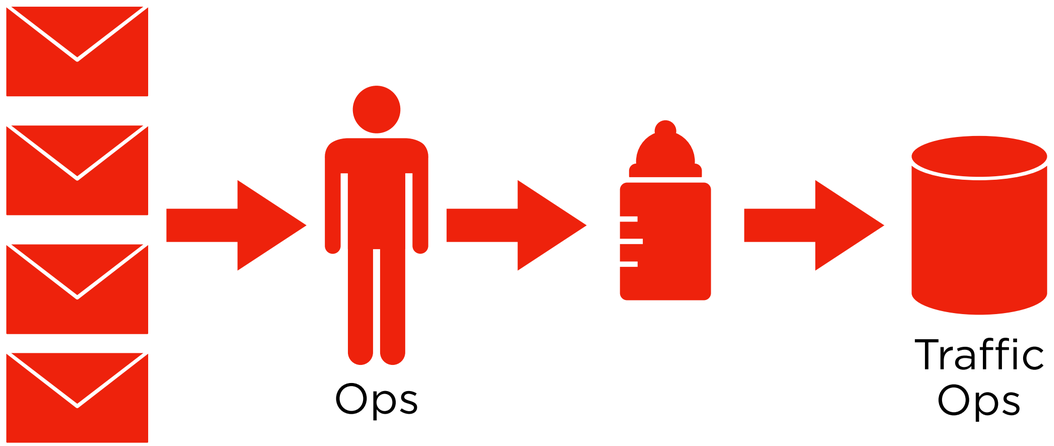
   A. Other users in their tenant

   B. Sub-tenants beneath their tenant

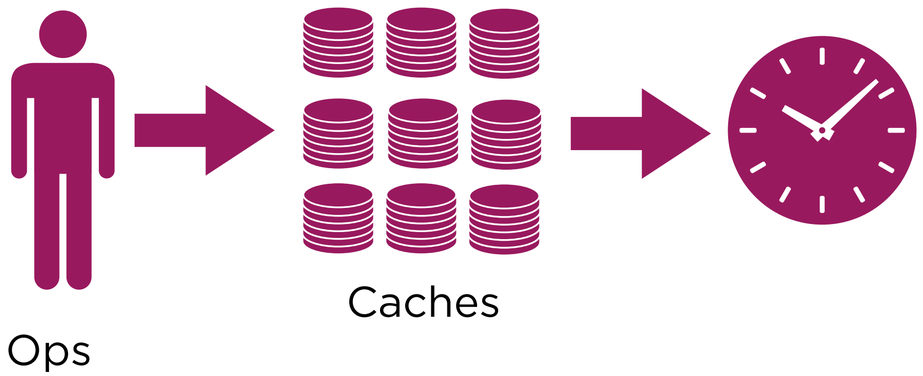
   C. Delivery services in their tenant

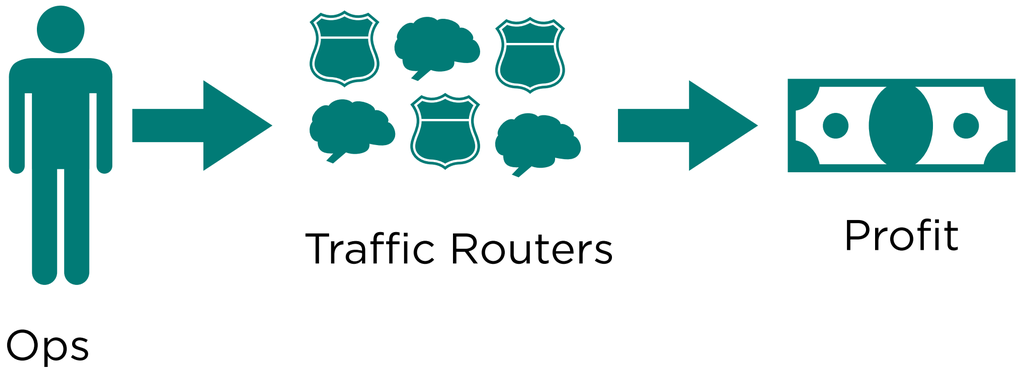
Current Workflow





Current Workflow





Current Workflow

Queue Updates

Pros:

- No partial changes from ops
- No accidental deployments
- Heavy sequence point

Cons:

- Manual, expensive
- Tooling is poor

CRConfig Snapshot

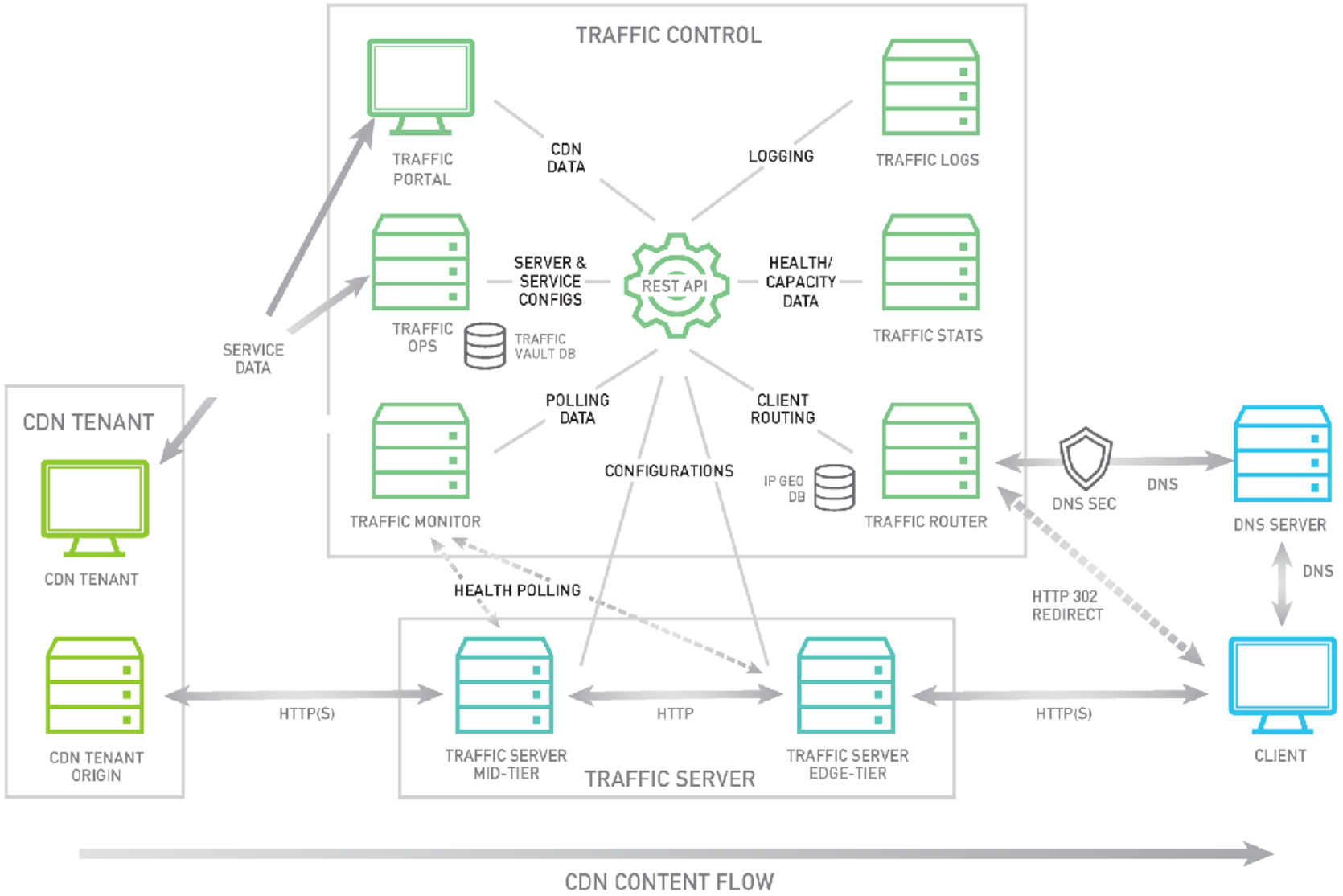
Pros:

- No partial changes from ops
- No accidental deployments

Cons:

- Scales horribly!
  - (8.9MB, 411,237 lines)
- Manual, expensive
- Tooling is poor

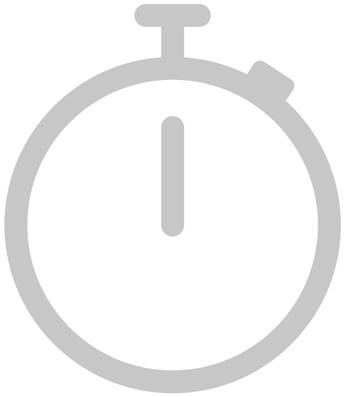
Current Architecture



Pull

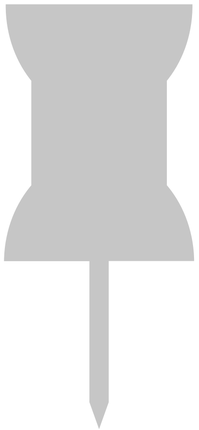
versus

Poll

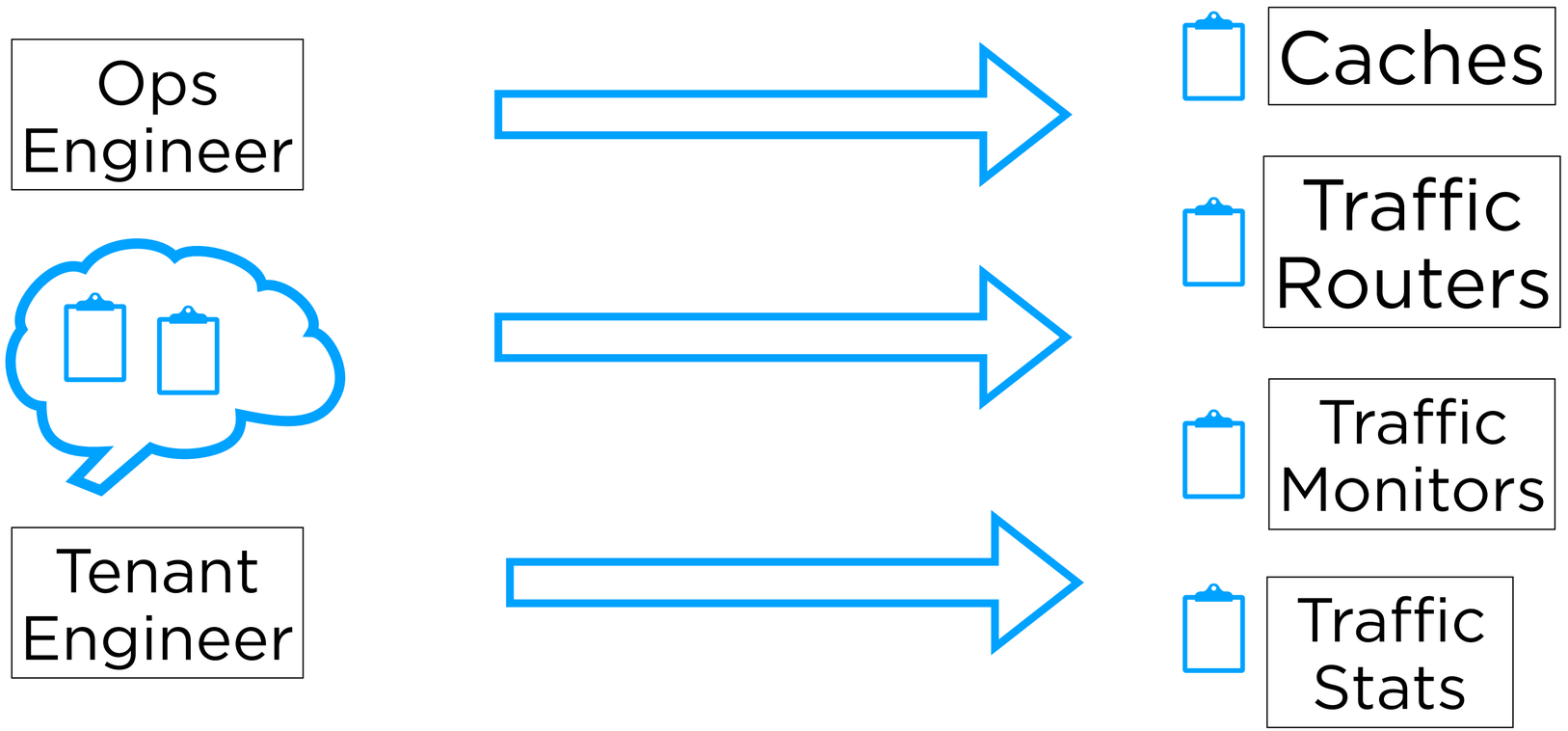


versus

Push



High-level goals



Distributed ChangeLog

Fancy set of diagrams go here.

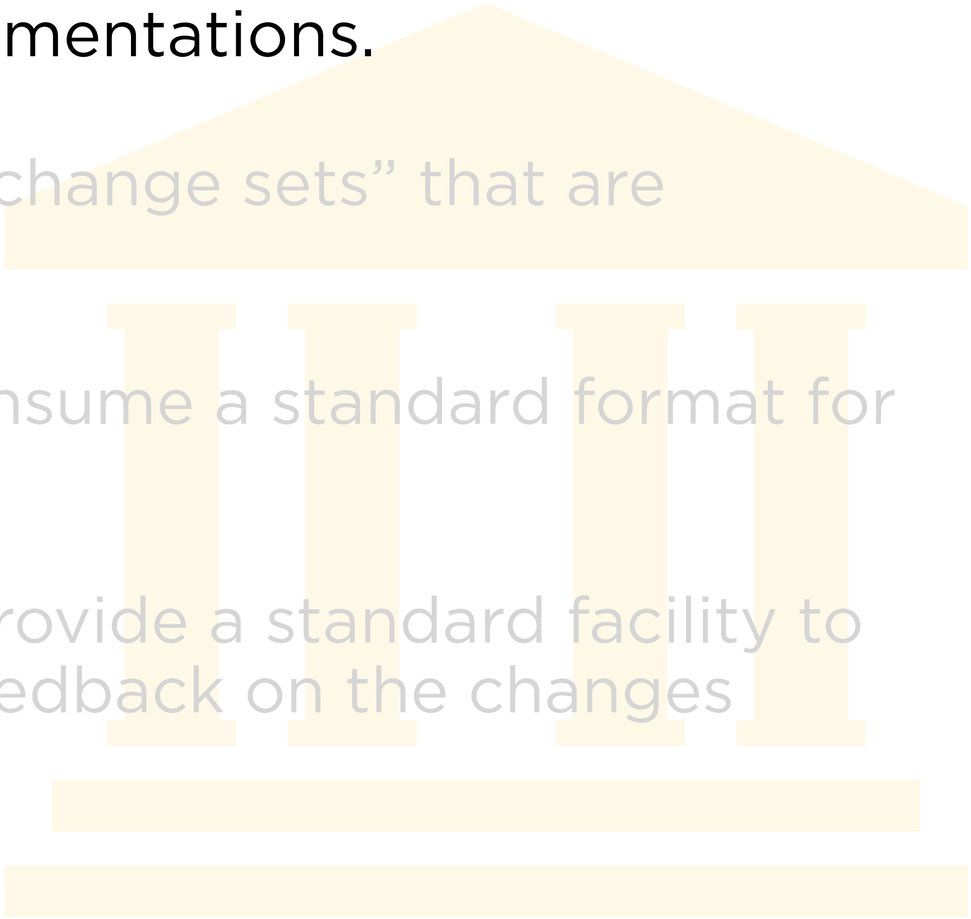
With Feedback

Another fancy set of diagrams go here.

Where does Error Handling go?

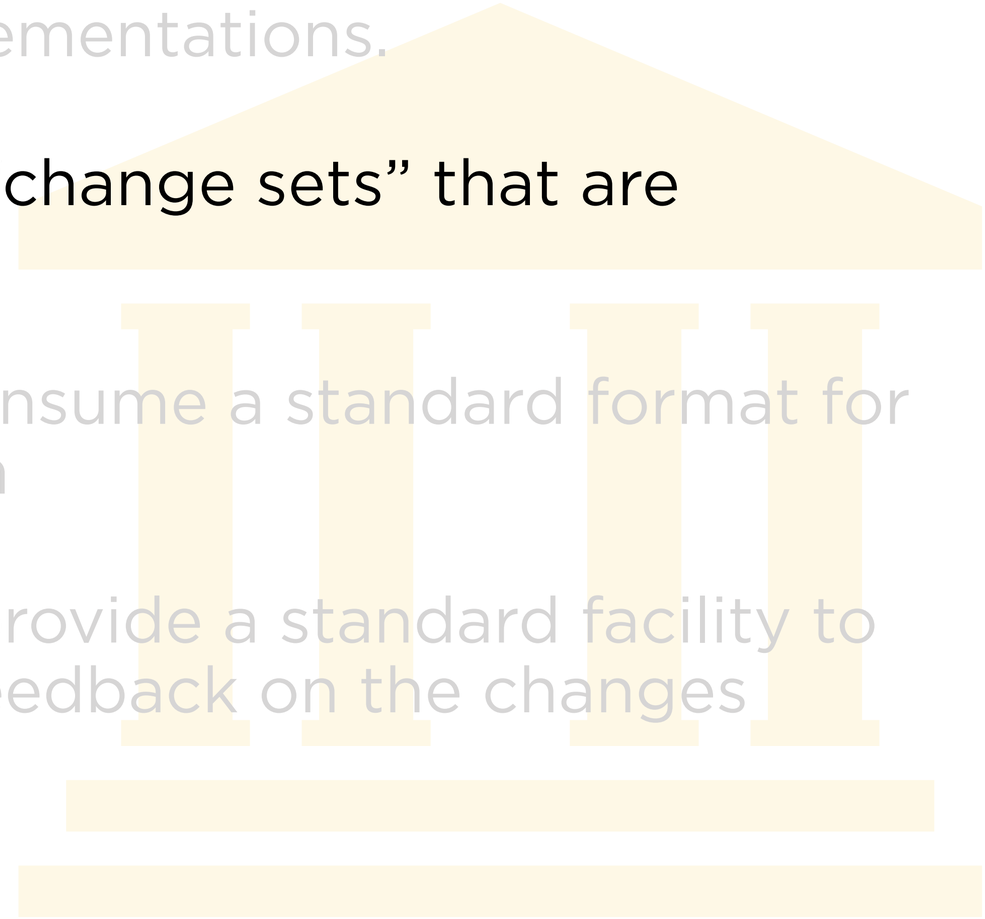
High-level design conclusions

1. Traffic Ops administers generic concepts, not software-specific implementations.
2. Traffic Ops generates “change sets” that are distributed
3. All components will consume a standard format for the same configuration
4. Each component will provide a standard facility to validate and provide feedback on the changes



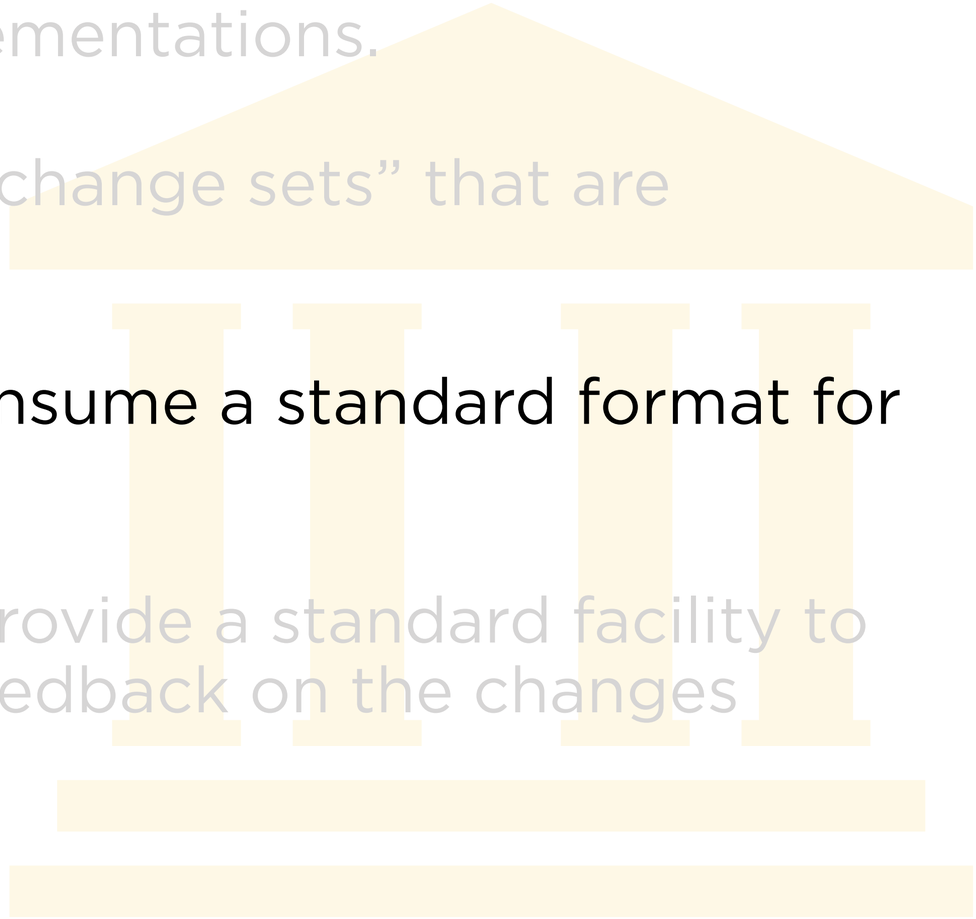
High-level design conclusions

1. Traffic Ops administers generic concepts, not software-specific implementations.
2. Traffic Ops generates “change sets” that are distributed
3. All components will consume a standard format for the same configuration
4. Each component will provide a standard facility to validate and provide feedback on the changes



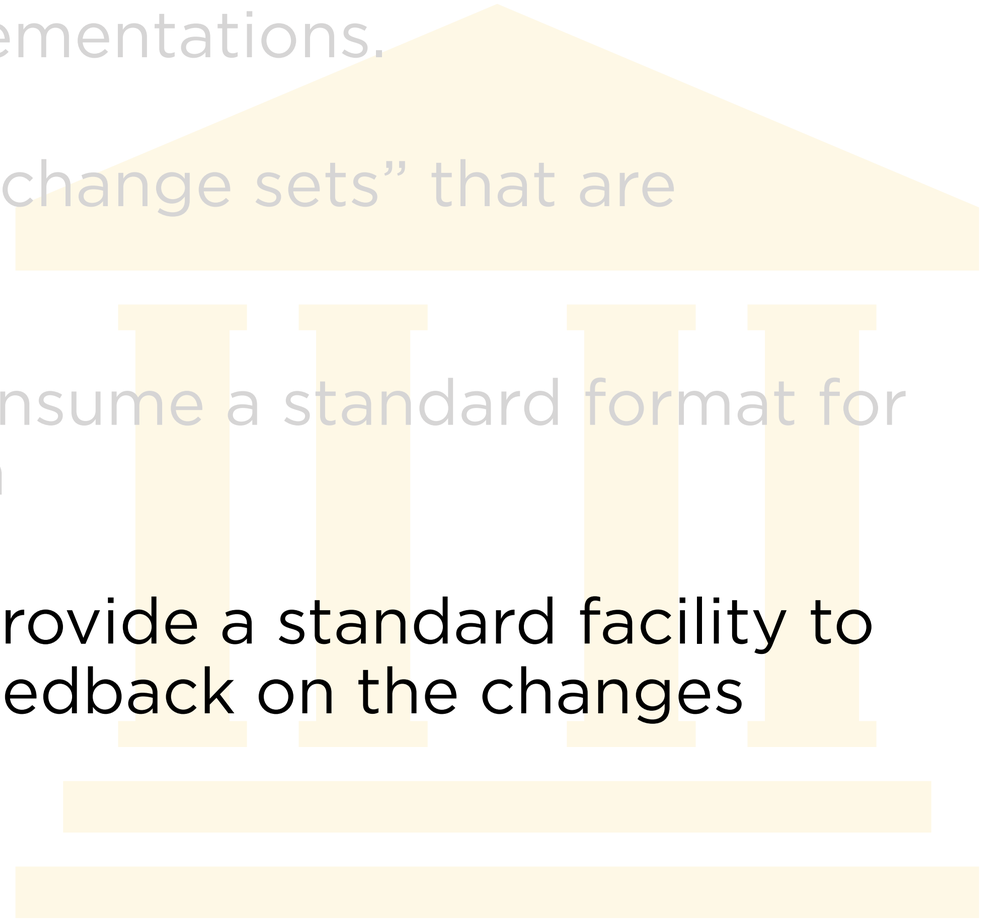
High-level design conclusions

1. Traffic Ops administers generic concepts, not software-specific implementations.
2. Traffic Ops generates “change sets” that are distributed
3. All components will consume a standard format for the same configuration
4. Each component will provide a standard facility to validate and provide feedback on the changes



High-level design conclusions

1. Traffic Ops administers generic concepts, not software-specific implementations.
2. Traffic Ops generates “change sets” that are distributed
3. All components will consume a standard format for the same configuration
4. Each component will provide a standard facility to validate and provide feedback on the changes



High-level design conclusions - continued

5. Traffic Ops continues to house the heaviest state. The state that exists in other components will lean towards ephemeral.
6. Zero manual intervention needed to achieve goals. Solution should “just work”.
7. Roll-back is not automated, roll-forward is.
8. For a given key, failure of one change should not affect
9. Time-to-running should feel immediate.

High-level design conclusions - continued

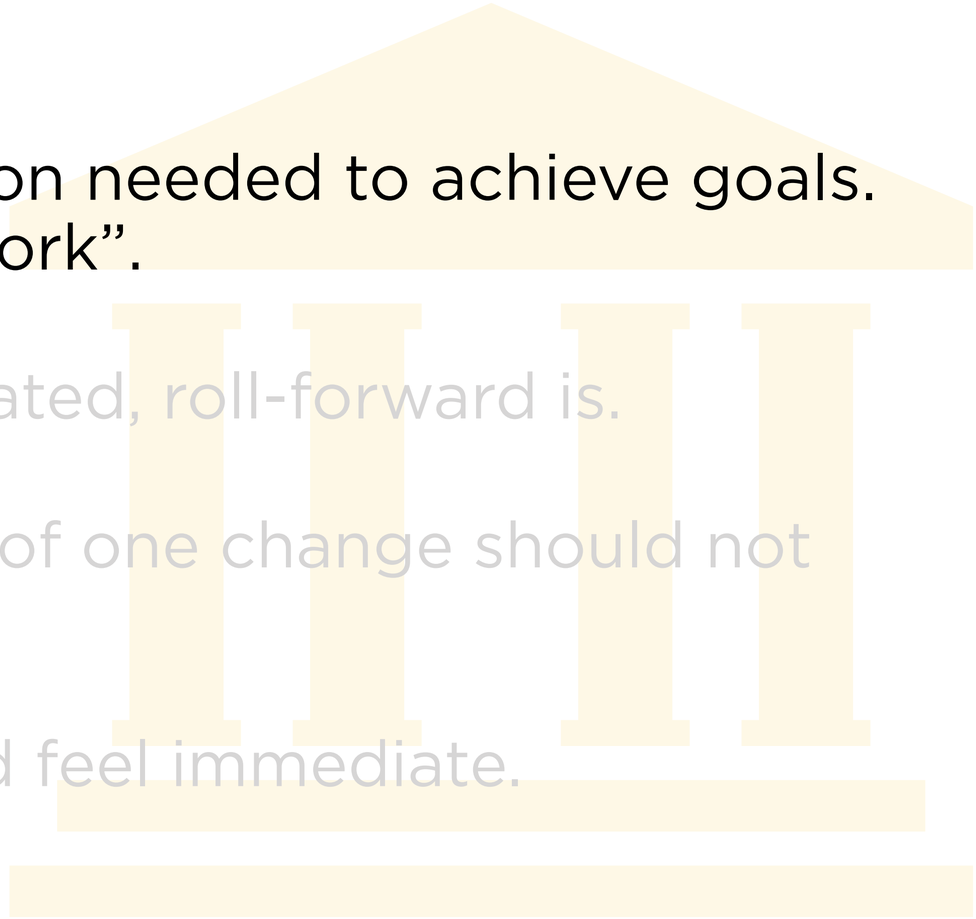
5. Traffic Ops continues to house the heaviest state. The state that exists in other components will lean towards ephemeral.

6. Zero manual intervention needed to achieve goals. Solution should “just work”.

7. Roll-back is not automated, roll-forward is.

8. For a given key, failure of one change should not affect

9. Time-to-running should feel immediate.



High-level design conclusions - continued

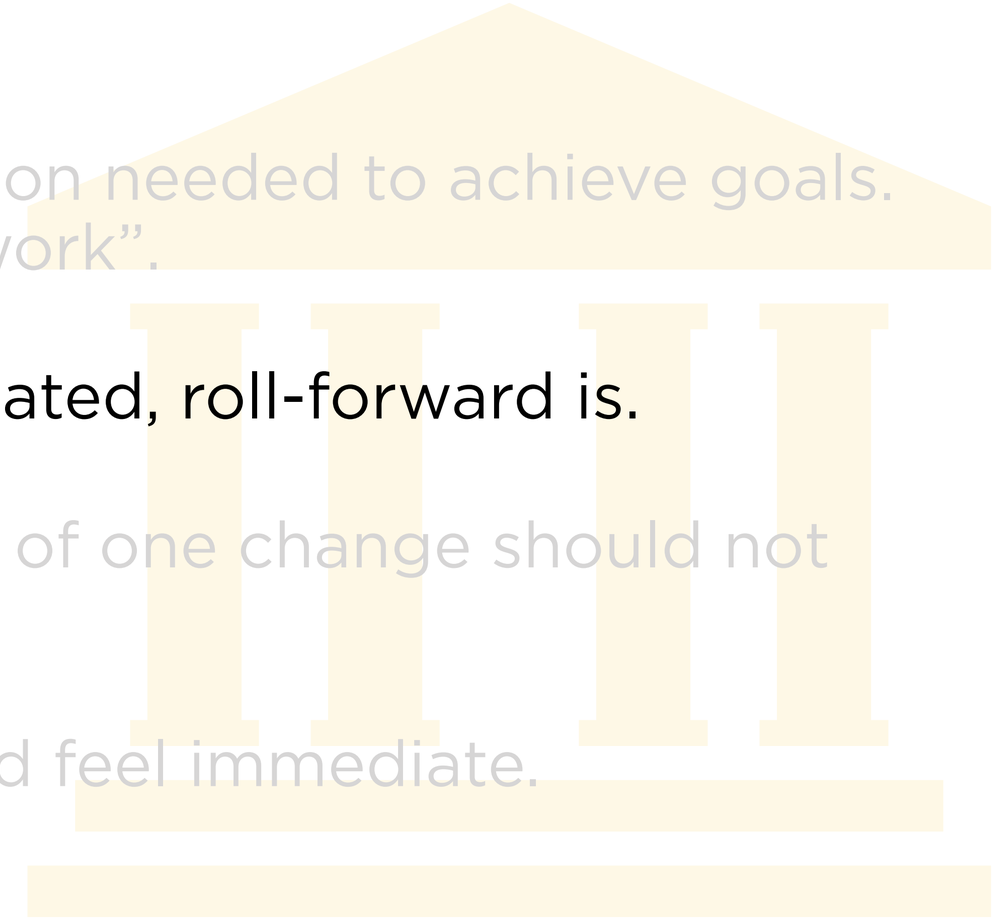
5. Traffic Ops continues to house the heaviest state. The state that exists in other components will lean towards ephemeral.

6. Zero manual intervention needed to achieve goals. Solution should “just work”.

7. Roll-back is not automated, roll-forward is.

8. For a given key, failure of one change should not affect

9. Time-to-running should feel immediate.



High-level design conclusions - continued

5. Traffic Ops continues to house the heaviest state. The state that exists in other components will lean towards ephemeral.

6. Zero manual intervention needed to achieve goals. Solution should “just work”.

7. Roll-back is not automated, roll-forward is.

8. For a given key, failure of one change should not affect future changes

9. Time-to-running should feel immediate.

High-level design conclusions - continued

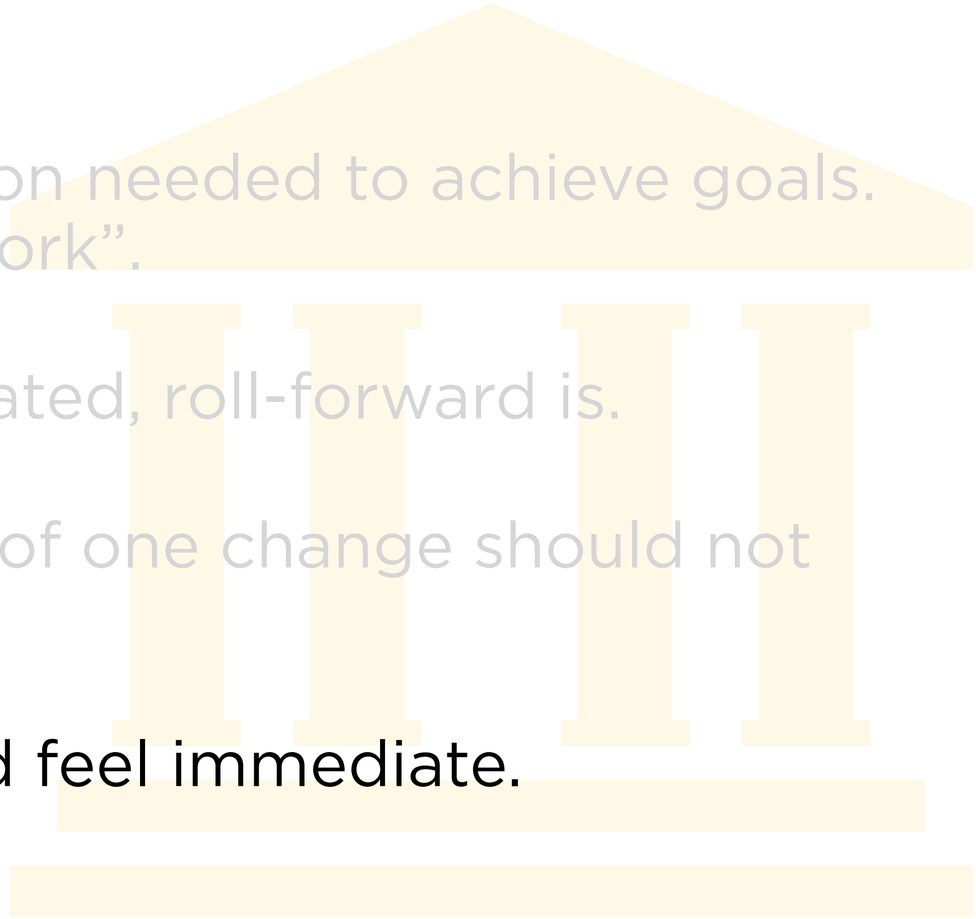
5. Traffic Ops continues to house the heaviest state. The state that exists in other components will lean towards ephemeral.

6. Zero manual intervention needed to achieve goals. Solution should “just work”.

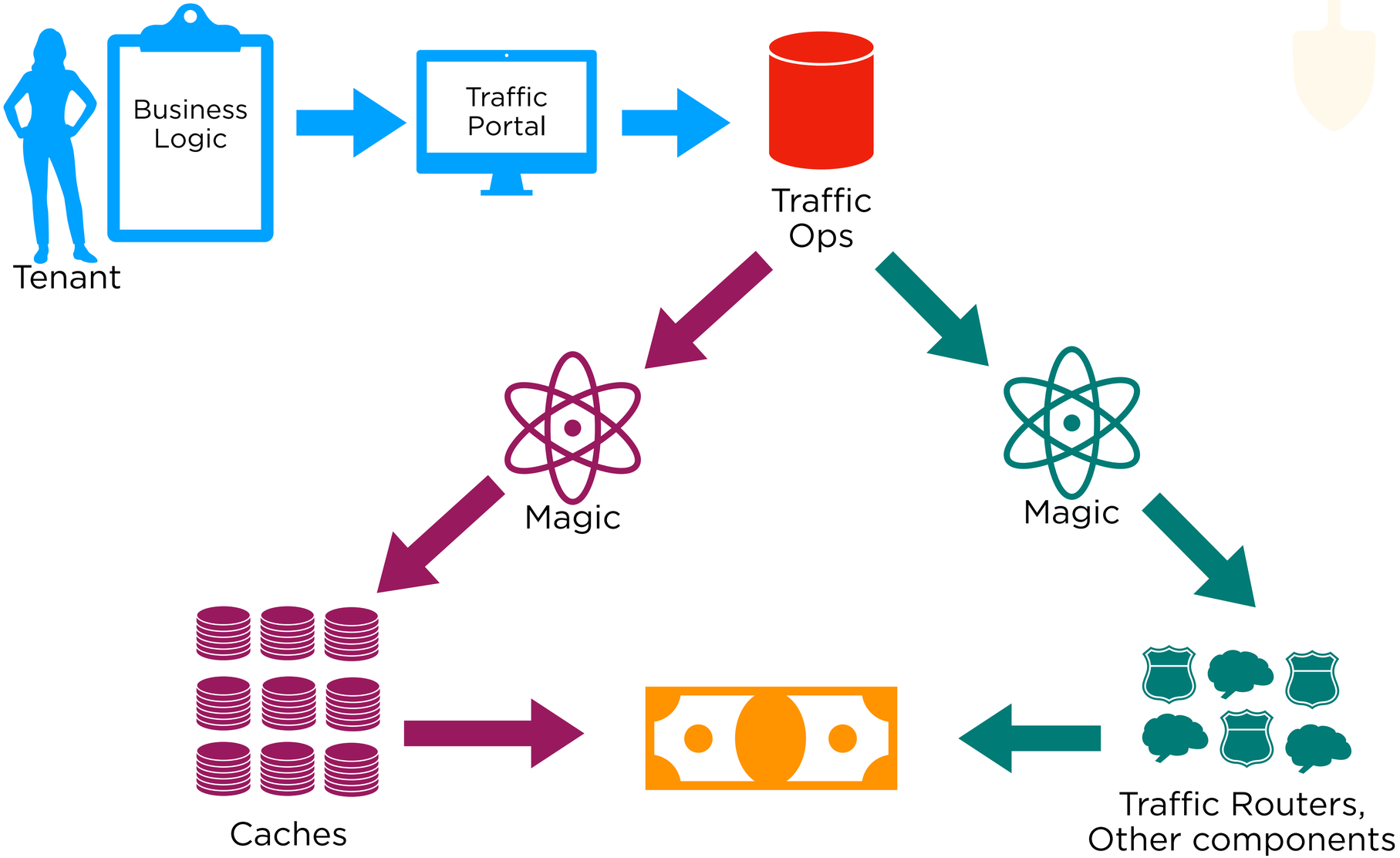
7. Roll-back is not automated, roll-forward is.

8. For a given key, failure of one change should not affect

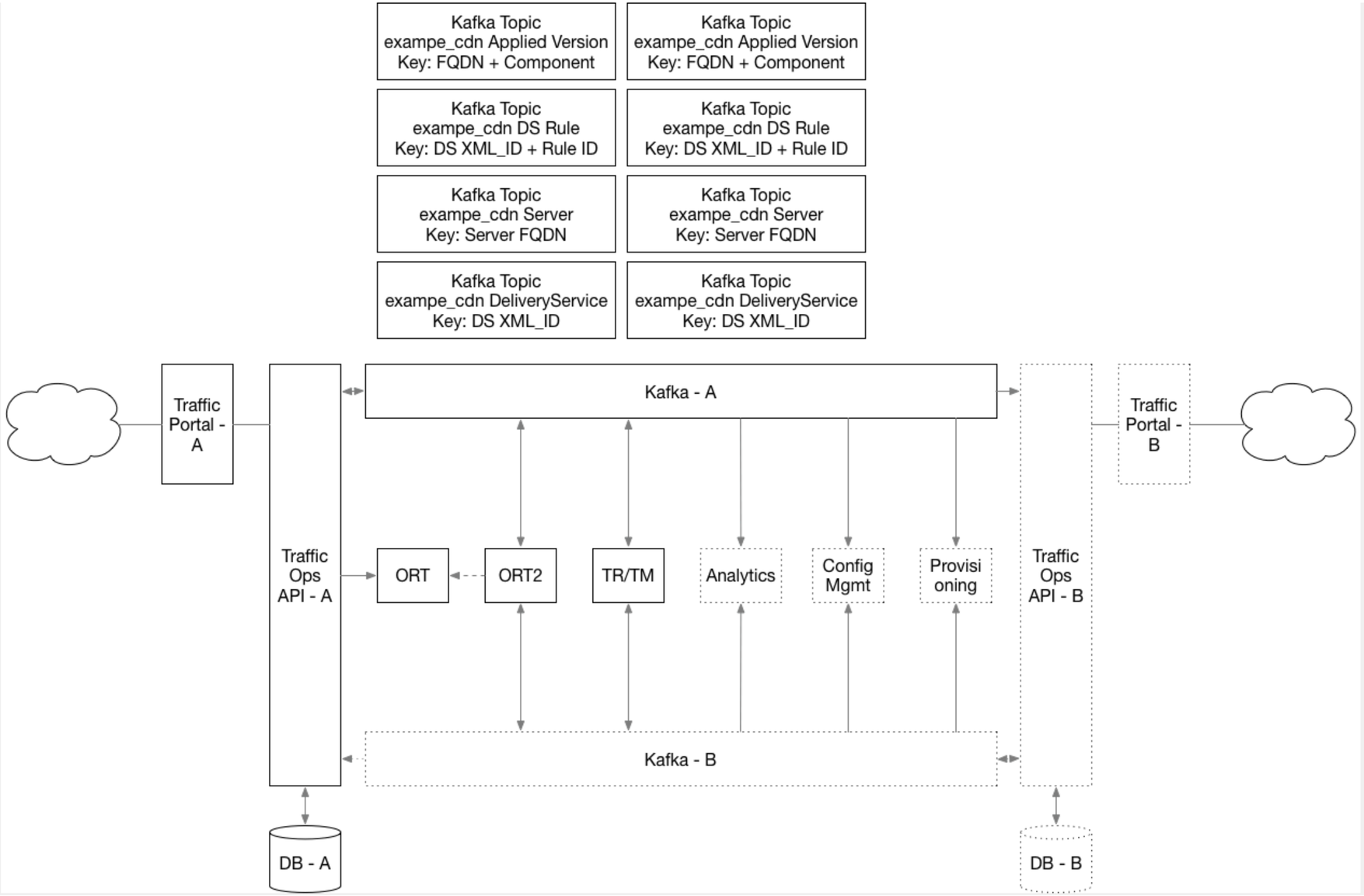
9. Time-to-running should feel immediate.



Future Workflow



Future Architecture



Kafka Topics & Keys

Topic (CDN name)	Key (scope.unique_identifier.sequence_point)
kabletown-cdn	ds.video-delivery-service.1508284754
kabletown-cdn	ds.images-delivery-service.1508285085
kabletown-cdn	cache.edge-cache-1-fqdn.1508284847
kabletown-cdn	cg.west-cache-group.1508284963
kabletown-cdn	user.markt.1508285139

Sequence Points & Feedback Loop

```
{ "ats": {
   "server": "6.2.2"
    },
   "system": {
     “inf.name": "bond0",
     "inf.speed": 20000,
     "proc.net.dev": "bond0: 0 0 0 0 0 0 0 0 0",
     "proc.loadavg": "2.28 2.42 2.23 2/1020 20303",
     "configReloadRequests": 150,
     "lastReloadRequest": 1508325772,
     "configReloads": 6,
     "lastReload": 1508277746,
     "astatsLoad": 1504111630,
    "something": "here"
    },
   “trafficControl” : {
    “configSequencePoints”: {
      “applied”: 1508335000”
      “rejected”: [
        1508331000,
        1508332000,
        1508333000,
        1508334000
      ]
    }
  }
}
```

JSON Changelog?

A

```
{
    "response": {
        "hostname": “edge1”,
        "profile": "EDGE1",
        "cachegroup": "cg1",
        "ipGateway": "10.1.0.1",
        "ipAddress": "10.1.0.2",
        "ipNetmask": "255.255.255.0",
        "interfaceMtu": 9000,
        }
}
```

B

```
{
    "response": {
        "hostname": “edge1",
     "profile": "EDGE1",
        "cachegroup": "cg1",
        "ipGateway": “10.1.0.1",
        "ipAddress": "10.1.0.3",
        "ipNetmask": "255.255.255.0",
        "interfaceMtu": 9000,
        }
}
```

Diff

```
{
    "response": {
        "ipAddress": "10.1.0.2",
        }
}
```

Properties File Changelog?

A

```
cache.edge1.profile.name EDGE1
cache.edge1.cachegroup cg1
cache.edge1.ipGateway “10.1.0.1”
cache.edge1.ipAddress “10.1.0.2”
cache.edge1.ipNetmask “255.255.255.0”
cache.edge1.interfaceMtu 9000
```

B

```
cache.edge1.profile.name EDGE1
cache.edge1.cachegroup cg1
cache.edge1.ipGateway “10.1.0.1”
cache.edge1.ipAddress “10.1.0.3”
cache.edge1.ipNetmask “255.255.255.0”
cache.edge1.interfaceMtu 9000
```

Diff

```
cache.edge1.ipAddress.1500000000 “10.1.0.2”
cache.edge1.ipAddress.1600000000 “10.1.0.3”
```

Delivery Service Add

```
envelope: {
  topic “kabletown-cdn"
  scope “ds”
  sequencePoint.scope.current 1500000000
  sequencePoint.scope.previous 1400000000
  sequencePoint.topic.current 1500000000
  sequencePoint.topic.previous 1450000000
}
response: {
  ds.video-delivery-service.ipAddress.hostregex.1500000000 “.*\.video-delivery-service\..*”
  ds.video-delivery-service.ipAddress.queryStringHandling.1500000000 “drop-at-edge”
  ds.video-delivery-service.ipAddress.maxDnsAnswers.1500000000 5
  ds.video-delivery-service.ipAddress.tlsEnabled.1500000000 true
  ds.video-delivery-service.ipAddress.active.1500000000 1
}
```

Kafka Topics & Keys

All components subscribe
to the topic in their CDN

Edit DS Use Case

- Question to group - user submits a change to their DS, change fails to apply to a component. What do we do?
  - Roll-back is not automated
  - DS gets marked as ‘un-validated’ in Traffic Ops?

New dependency!

- Traffic Configurator (Kafka)! (kidding)
- Shoot. The last thing we need is another dependency to get your CDN working.
- ORT (or replacement) will still be able to work. (Non-self-service mode should still be a thing.)

Sounds like PubSub

Why not actually just use PubSub?

- Eh, could. Maybe.
- Existing implementations seem to fall short.
- No momentum?
- Like lots of things, the current implementations seem to fall short. This is important enough to us to roll our own.
  - The feedback loop is crucial to this being reliable.

Super Advanced Config

Will still need to be changed on a DS manually, by a trusted professional

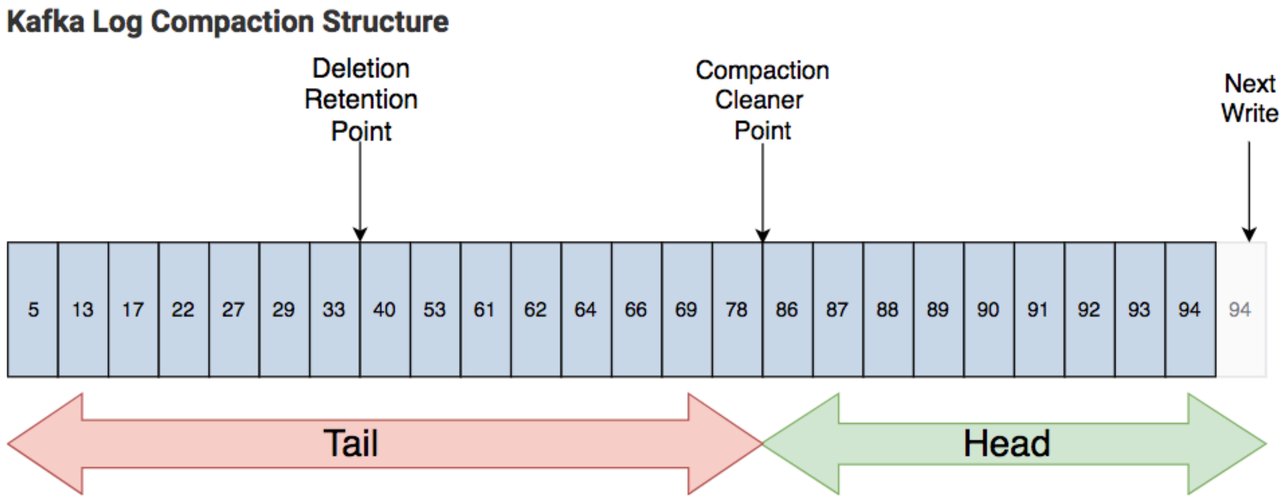
What about the bootstrap case?

Kafka log compaction to the rescue!

Kafka Log Compaction Structure

With a *compacted log*, the log has head and tail. The head of the compacted log is identical to a traditional Kafka log. New records get appended to the end of the head.

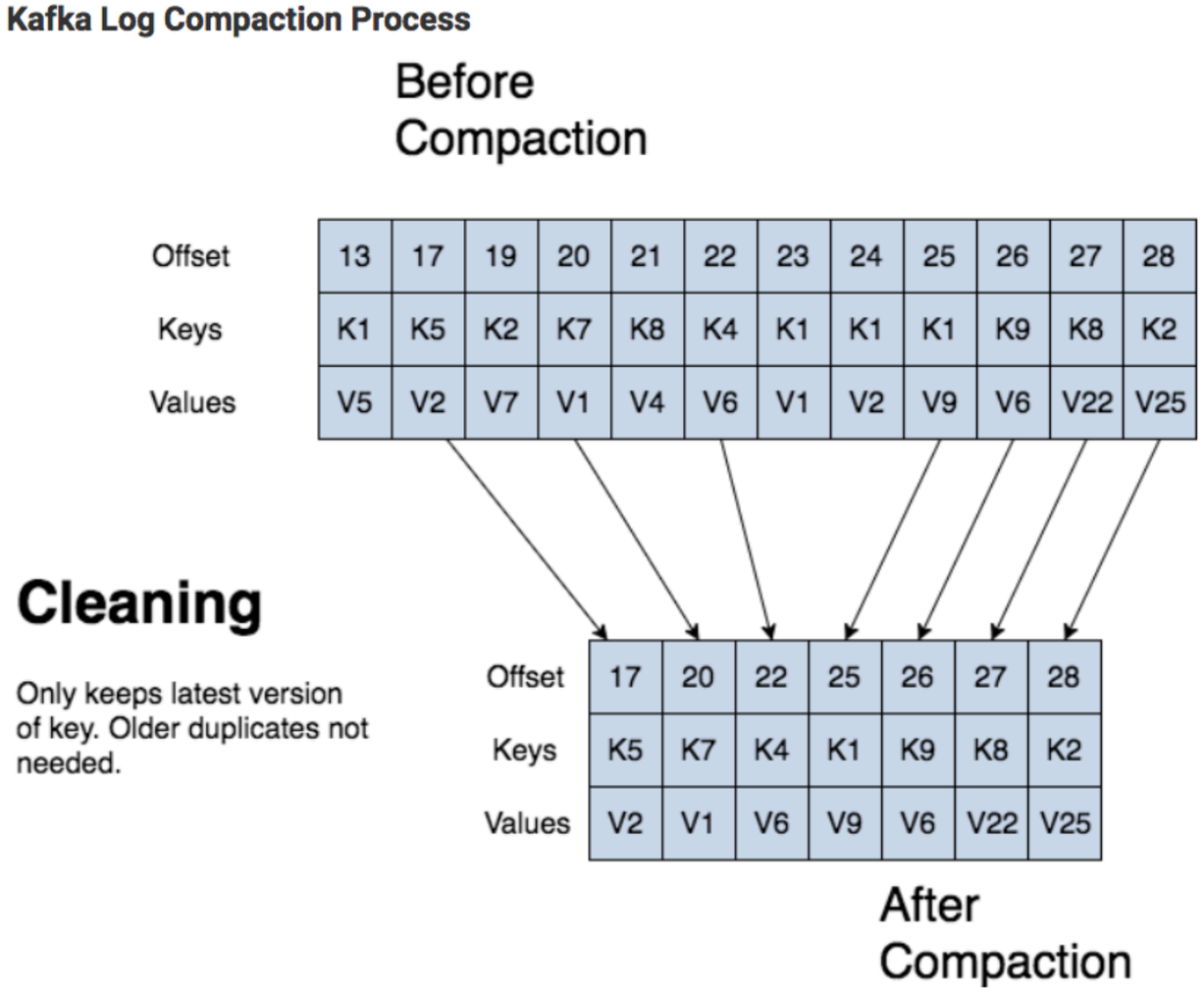
All log compaction works at the tail of the log. Only the tail gets compacted. Records in the tail of the log retain their original offset when written after being rewritten with *compaction cleanup*.



http://cloudurable.com/blog/kafka-architecture-log-compaction/index.html

What about the bootstrap case?

Kafka log compaction to the rescue!



LOE per component

Self-service 0.1

```
traffic_ops=# \d deliveryservice;
                                    Table "public.deliveryservice"
           Column            |           Type           |                          Modifiers
-----------------------------+--------------------------+-------------------------------------------------------------
 id                          | bigint                   | not null default nextval('deliveryservice_id_seq'::regclass)
 xml_id                      | text                     | not null
 active                      | boolean                  | not null default false
 dscp                        | bigint                   | not null
 signed                      | boolean                  | default false
 validated                   | boolean                  | default false
 qstring_ignore              | smallint                 |
 geo_limit                   | smallint                 | default '0'::smallint
 http_bypass_fqdn            | text                     |
 dns_bypass_ip               | text                     |
 dns_bypass_ip6              | text                     |
 dns_bypass_ttl              | bigint                   |
 org_server_fqdn             | text                     |
 type                        | bigint                   | not null
 profile                     | bigint                   |
 cdn_id                      | bigint                   | not null
 ccr_dns_ttl                 | bigint                   |
 global_max_mbps             | bigint                   |
 global_max_tps              | bigint                   |
 long_desc                   | text                     |
 long_desc_1                 | text                     |
 long_desc_2                 | text                     |
 max_dns_answers             | bigint                   | default '0'::bigint
 info_url                    | text                     |
 miss_lat                    | numeric                  |
 miss_long                   | numeric                  |
 check_path                  | text                     |
 last_updated                | timestamp with time zone | default now()
 protocol                    | smallint                 | default '0'::smallint
 ssl_key_version             | bigint                   | default '0'::bigint
 ipv6_routing_enabled        | boolean                  | default false
 range_request_handling      | smallint                 | default '0'::smallint
 edge_header_rewrite         | text                     |
 origin_shield               | text                     |
 mid_header_rewrite          | text                     |
 regex_remap                 | text                     |
```

Opening questions

1. What is self-service?
2. Who thinks Self-Service needs to be a priority for Traffic Control?
3. What would folks like to discuss in this talk?
4. What would folks like to get out of this session?
    1. I would like to get a loose consensus on the direction — we don’t get together often (summits, hangouts, etc), so we need to capitalize