TeRI over DRiP

Modern Working Group
IETF100
Chris Wendt

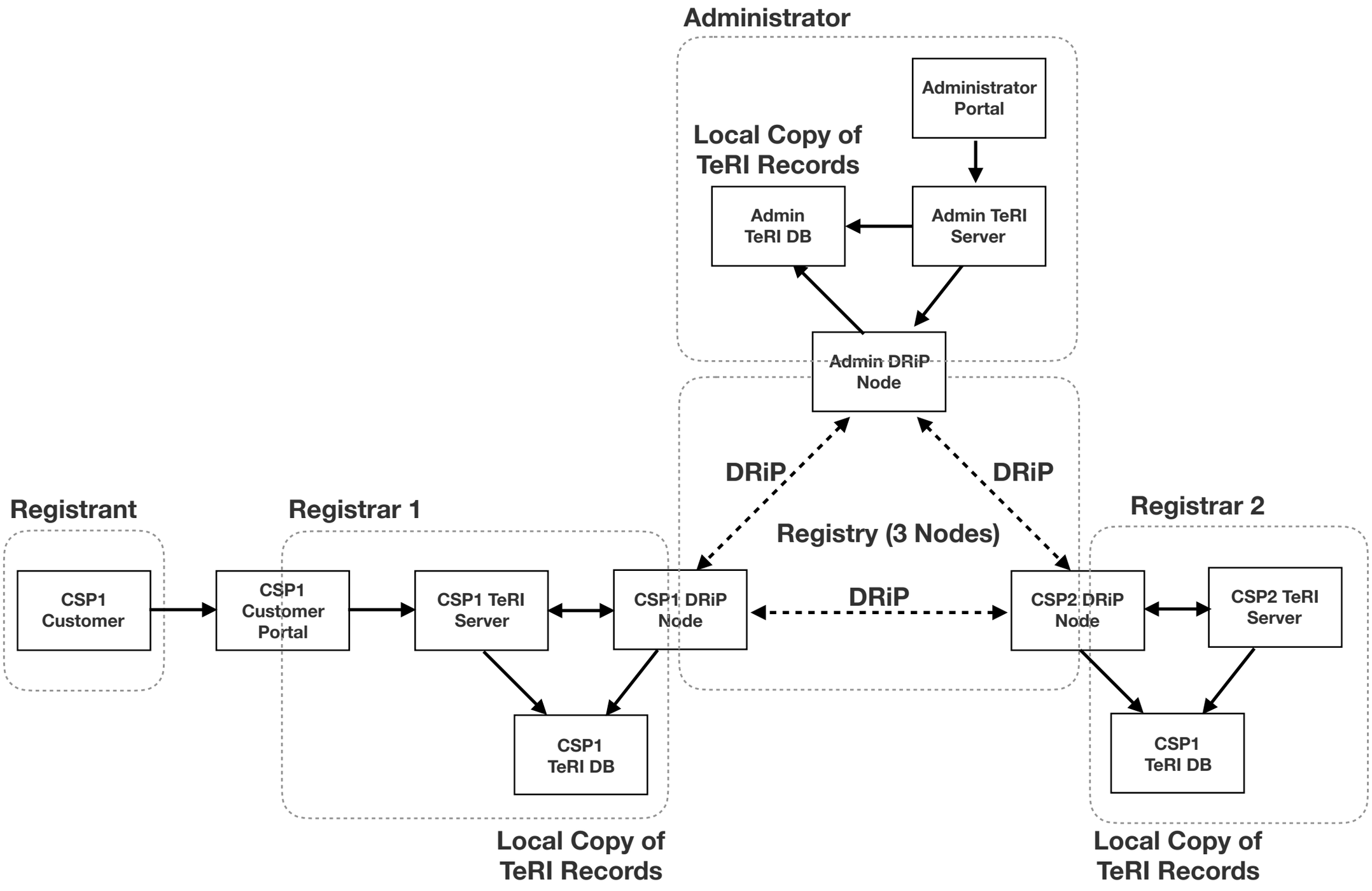
Overview

- TeRI is a framework and data model for managing telephone number resources.

- DRiP is a HTTP based protocol for transporting JSON key value pair data objects between interconnected nodes across a network

- This is an overview of how TeRI and DRiP can be used together for a telephone number resource management framework across a distributed set of consumers and users.

TeRI/DRiP Actors, Example 3 Node Architecture



TeRI Request - Response

- I haven't found a good way to map this specifically to interaction model for all cases

- For example: a query to give me all of the assigned numbers for 1215555 is like an HTTP GET with a 200OK response, this is straight forward.

- But an assignment operation where you might have a CSP and Admin doing two different operations for a request, a policy decision, and then a validation doesn't necessarily map to just a request-response

TeRI/registry interaction model

- Assumption: Transactions are registry based
- Registrar to Administrator or Registrar to Registrar(s) transactions are based on Creating, Updating TeRI records in Registry
- Permissions to create and update records and role validation should all be based on signatures over created TeRI data or added/updated TeRI data associated with the appropriate credentials and role implied by those credentials

TeRI/registry interaction model 2

- Since transactions are registry based, there should be two interaction models
- One sided: Actor performs record creation/update/delete -> success/fail
- Two or perhaps N sided: Actor 1 performs record creation/update/delete -> success/fail, Actor 2 receives this new record and reacts by performing record update (not create and likely not delete) ->success/fail

Example TeRI/registry interaction

- Allocate 10k block (two way): CSP1 requests 10k block allocation -> Admin confirms 10k block allocation with it's authoritative signature.

- Assign TN (one way): CSP1 assigns TN in owned TN (signatures should match CSP1 so valid)

- Assign TN Error condition (one way): CSP1 assigns TN in CSP2 TN block (signature certificates do not match, so bogus entry)

TeRI Acquisition Operations

- Request for TN or TN Block from CSP and authorization from Administrator
- Two steps of pending acquisition, and acquisition authorization into the registry
- In terms of CRUD or DB, this is “Create”
- In terms of HTTP, this is PUT verb

TeRI Management Operations

- Request to modify a record
- This could be One step or Two step depending on how self policing we want the registry to be
  - CSP1 modifies a TN information in it's own TN block allocation, or does Admin need to validate that for every transaction
- In terms of CRUD or DB, this is "Update"
- In terms of HTTP, this is POST verb

TeRI Retrieval Operations

- Request to get a record
- This would always be a one step operation without any validation
- Assume all records are available to all? If no, impact to distributed registry because data gets distributed to all?
- In terms of CRUD or DB, this is “Retrieve”
- In terms of HTTP, this is GET verb

TeRI->DRiP dependent operations

- DRiP includes a voting phase and a commit phase
- DRiP currently includes the payload in voting phase
- We could take advantage of this fact for two step transactions and use the voting phase to validate operations and policy before registry entries are even committed to registry
- For example, error condition handling before committing
- Do we want to make TeRI/DRiP dependent or independent or support both (not completely sure how that would work)

Async vs Sync TeRI->DRiP operations

- Should we consider async operations/API? Push/ callback/promise
- Distributed registry will take “more” time for operations to complete, but that is relative (e.g. seconds vs milliseconds)
- Does it matter? Probably Async vs Sync can be implementation specific based on Transport and other criteria, but should consider this in either case.