

# Future of CTS Scale

Jonathan Scudder / Robert Wapshott

Future of the CTS UnSummit Notes:

Opportunity to present current thoughts on the direction we will take the CTS in to achieve higher orders of scale.

## 1) Reworking the OAuth2 data model

Examining the OAuth2 data model and the way it is persisted in the CTS we conclude that there are a number of optimisations we can make to this storage format to enable higher performance with a lower cost of ownership for this use case. Combining all tokens for an OAuth2 Client into a single token allows us to change the nature of the database operations from Add/Delete to Modify operations thus saving further on indexing costs.

## 2) General token storage improvements (including specific schemas)

In addition to the OAuth2 restructure we note that there are more improvements we can make to the overall token schema used for the CTS. Changing from a generic schema to a token specific schema will allow easier debugging in the field and focus our indexes to ensure that the product ships with indexes which require less specific tuning.

## 3) New reaper approach

The CTS Reaper is a component which cleans up expired tokens within the DJ persistence store. This feature can be further improved by moving this task to the DJ persistence store, effectively introducing a TTL function to the database. We believe the biggest benefit from this will be the reduced load on replication from not having to replicate delete tokens around the replication cluster.

## 4) Improved ease of management, particularly rolling upgrades

Each of the features of the CTS has some administrative complexity associated with it. The feedback we receive from customers indicates that efforts to simplify this will be beneficial. We can address this by exploring options for automatic configuration and discovery features to simplify CTS configuration and administration.

## 5) Data centre affinity

In global deployments, maintaining a consistent service is a continuous challenge. In order to help this increasingly common use case we can look to include data centre routing information into the tokens to allow for AM to route the request to the original data centre where the token was stored. This might come at the cost of increased latency per request but will save on the requirement for tokens to be quickly replicated around the world. This can be backed up with a fall-back behaviour of checking the local store as required.