



# Public REST API and Developer Portal Roadmap

08/30/2023



# Forward-Looking Statement

Statements regarding our product development initiatives, including new products and future product upgrades, updates, or enhancements, represent our current intentions but may be modified, delayed, or abandoned without prior notice, and there is no assurance that such offering, upgrades, updates, or functionality will become available unless and until they have been made generally available to our customers.

# Public APIs

## Completed

- (3900.69) Read Rubric Evaluations
- (3900.71) Read Gradebook Logs
- (3900.74):
  - Expand Rubric associations in Gradebook column endpoints
  - REST APIs to REST API to track the origin of courses
  - REST APIs to REST API to modify formulas for enhanced grading control

## Next (Oct-Dec)

- Gradebook: add dates to results and searches
- Courses include copy\_from\_identifier in response
- LTI context and resource history
- Calendar: Enable non-3LO calendar management
- Institutional Hierarchy: Move child nodes
- Course Content: Implement resource handler gaps. E.g.: blankpage, toolinks, etc
- Add User secondary Ids

## Later (release TBD)

- Original Discussion APIs
- GET Logs
- Create/Update/Delete Content Collection Resource
- Content Collection - Ability to upload zip files
- Assessment APIs
- And more...

# Public APIs: *and more... (not prioritized)*

API	Description
Activity Stream	Provide ability to manage Activity Stream content. (CRUD)
Attendance API	General refactoring of Service APIs
Batch API requests	Enable submitting multiple 'like' operations in a single request.
Course Quotas	Provide APIs for Course Quota CRU
Data Sources	Improve searches
Group Grading	Group attempt grading CRUD
Multiple Search Criteria	Add ability to use multiple search criteria
OAuth: Tokens	Ability to revoke and force refresh OAuth tokens
System Endpoint	Add System stats to System endpoint (quotas: disk, DB, etc)
Terms	Search option on Name
UltraStatus management	Course Ultra Status management - set ultra status on courses
Users	Add Add user emails to the list of secondary Ids
Users	Allow User searches based on firstname+lastname; Add studentId to list of User secondary IDs

# Developer Portal

## Completed

- Announcement Panel
- LTI Dynamic Registration
- Redis Upgrade

## Next (Oct-Dec)

- Audit Trail
- Token Bucket Request Limiting...

## Later (release TBD)

- API Usage Reporting
- JWT for REST Apps
- Restrict 'cross-domain' application registration
- Restrict deployment of non-production applications
- Improve OpenAPI docs
- Improve integration deployment for administrators...

# New Token Bucket Rate\_Limiting Model

Token Bucket stores some number of tokens not more than bucket capacity<sup>1</sup>. Tokens are requested before a request may be processed. If there are enough tokens, bucket capacity is reduced by one and the request processed. Otherwise, the request is managed per a defined throttling model.

A Token Bucket is refilled with N tokens at a defined time (BFRT)<sup>2</sup>. This controls number of allowed actions over time. This allows for handling traffic bursts, depending on the number of available tokens.

When the Token Bucket is depleted, experiences sustained requests beyond the bucket's capacity, or exceeds the allotted contractual request count, limiting enters an overage/throttling phase.<sup>3</sup> During the overage/throttling phase requests are conditionally processed in a manner dictated by the volume and duration of the burst/usage. Developers and stakeholders are informed of throttling cases for proactive adjustment/outreach.

***In all cases the integration continues to function albeit with potential perceived performance degradation in the latter case.***

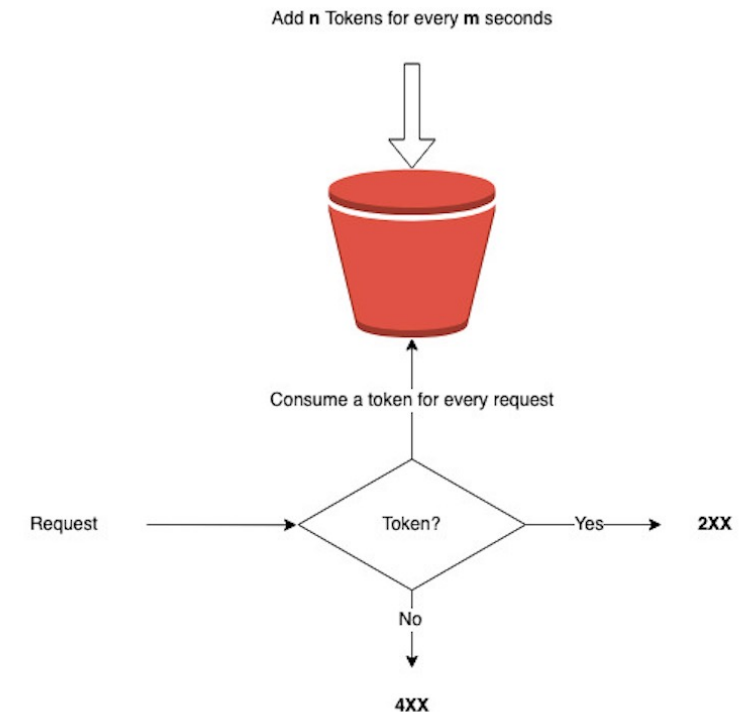
<sup>1</sup> bucket capacity corresponds to existing limits – e.g. 10K for primary groups.

<sup>2</sup> BFRT is the refill rate and corresponds to the current refresh rate, however in our new model T is in milli-seconds vs 24 hours.

<sup>3</sup> throttling phase slows the request processing queue and provides an opportunity for collecting fees re API use.

Note: we will continue to support portal admin overrides of settings on individual groups.

Token Bucket Algorithm for Rate Limit



# New Token Bucket Rate\_Limiting Model

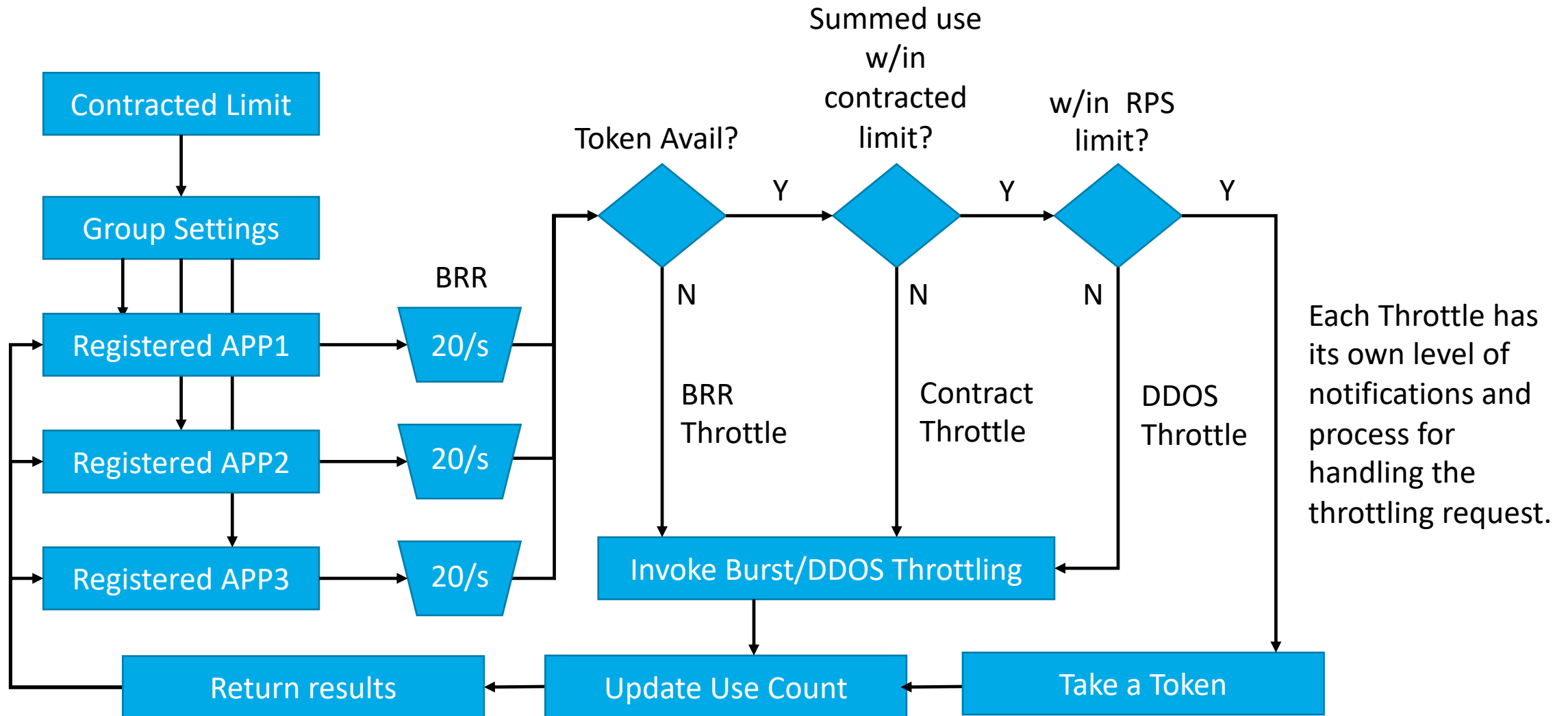
## Guiding principles behind Token Bucket Rate Limiting:

1. Integrations never stop due to rate-limiting
2. Contracted Limits set the number of allowed requests during a specified time period – *per Group and shared across applications.*
3. Contracted limits set the point at which integrations enter throttling based on exceeding usage during the specified Contracted period. E.g, 24 hrs.
4. Throttling – throttling *may* occur at three points in an application life cycle:
  1. When the bucket capacity is exceeded – this is a mild throttling with notification to developer/group owner.
  2. When the contracted-limit is exceeded – this is a mild throttling, with notification to developer/group owner. This is where we would collect overage fees.
  3. When all bucket capacity, soft-limit, and hard-limits are exceeded. This typically occurs from a DDOS attack, runaway app, or intentional overuse and is a degrading throttling. Notification is sent to both Portal admins and developer. Note we never completely stop processing requests – the integration will display performance degradation.

## Scenario:

1. Time period: 31 days
  2. Your bucket is 10K/day
  3. Your contracted limit is 310K/month
  4. Your contracted hard limit is 500K/month
  5. The bucket replenish rate (BRR) is 20/second
  6. We only allow max of 200 requests per second under all conditions (RPS)
- If you stay within your BRR, you are fine
  - If you stay within your BRR while under contracted limits, you are fine.
  - In all cases throttling is invoked when:
    - you exceed your BRR while under contracted-limits
    - you exceed your contracted limit
    - you exceed our RPS

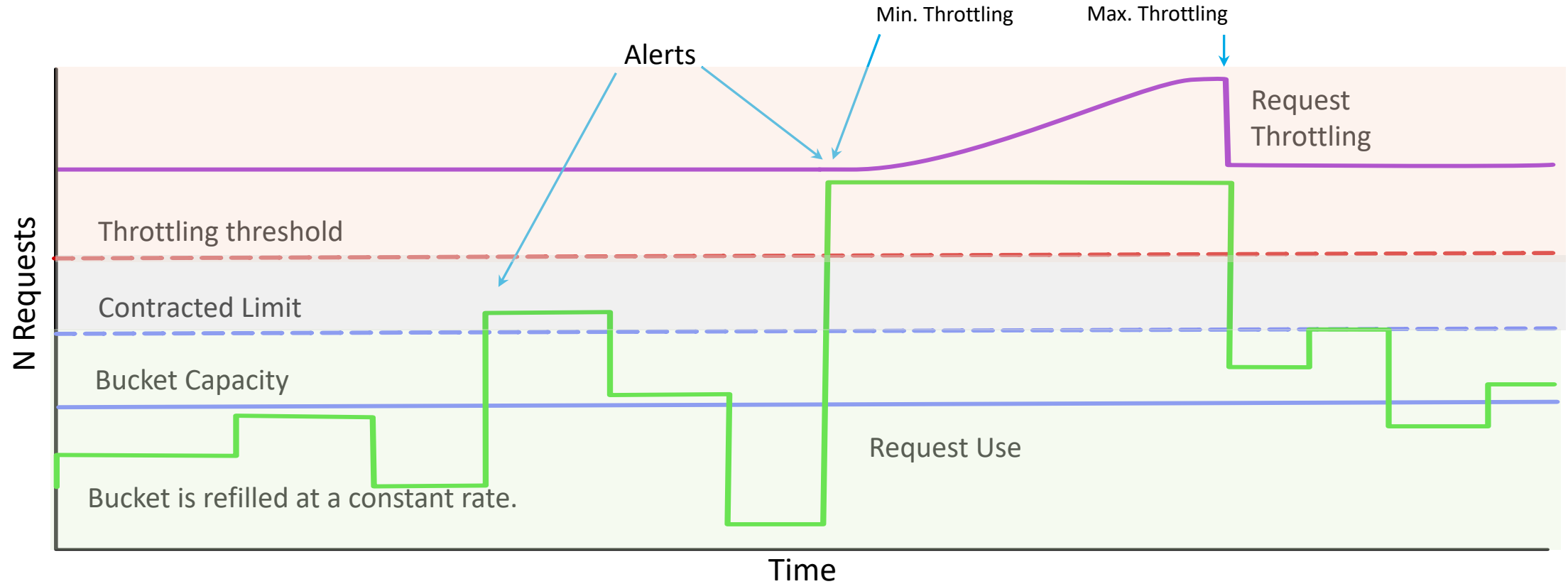
# New Token Bucket Rate\_Limiting Model: A Visual



All non-auth result headers contain limit information: <https://www.ietf.org/archive/id/draft-polli-ratelimit-headers-02.html>

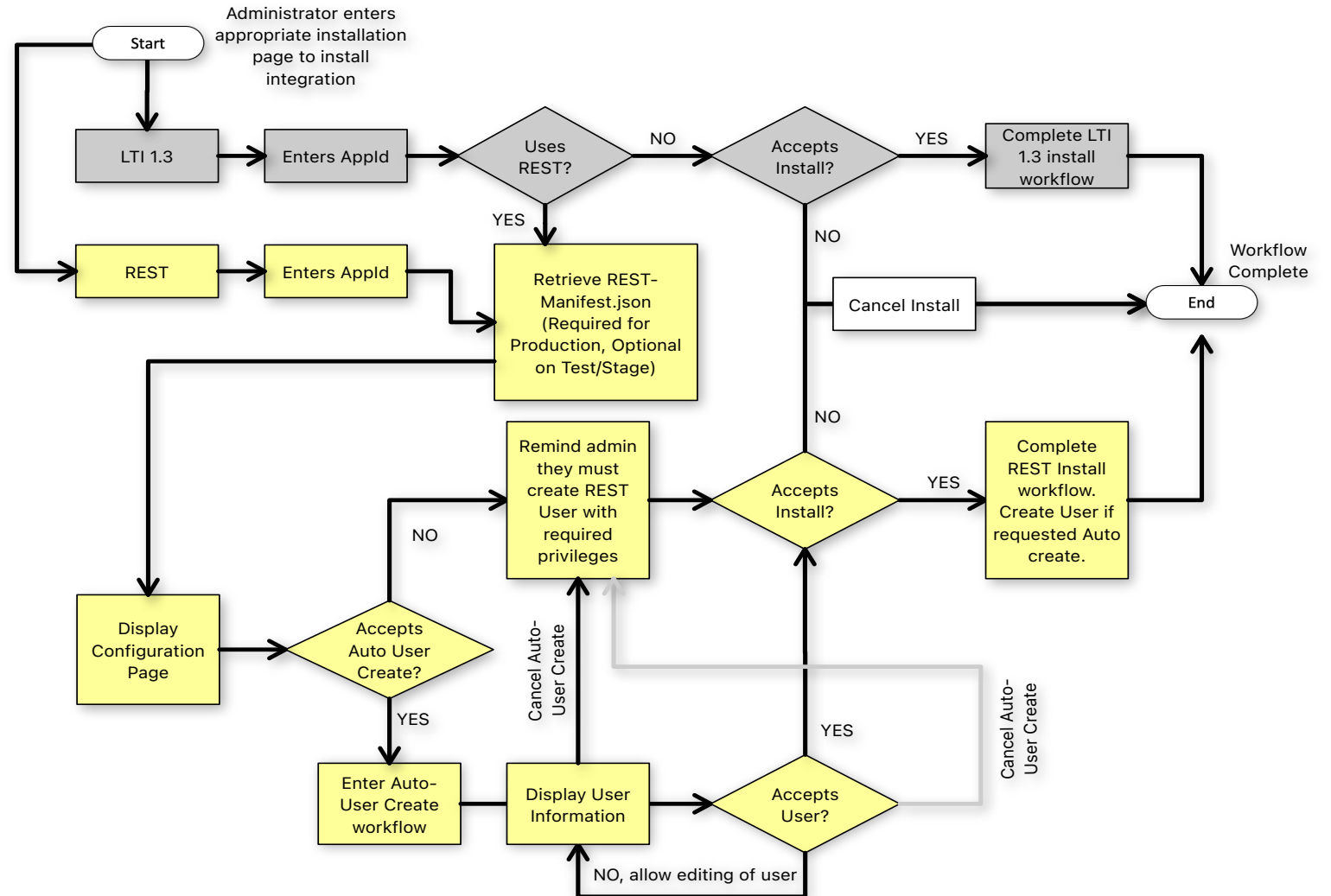


# New Token Bucket Rate\_Limiting Model in action...



# Improve integration deployment for administrators

1. Remove requirement for Admins to create integration users and roles.
  1. Admins may choose to create user and role based on #2 and #3.
2. Provide integration information to Admins ala previous integration installation processes.
3. Require Developers to provide an integration manifest for Production release.





anthology  
Advancing Knowledge