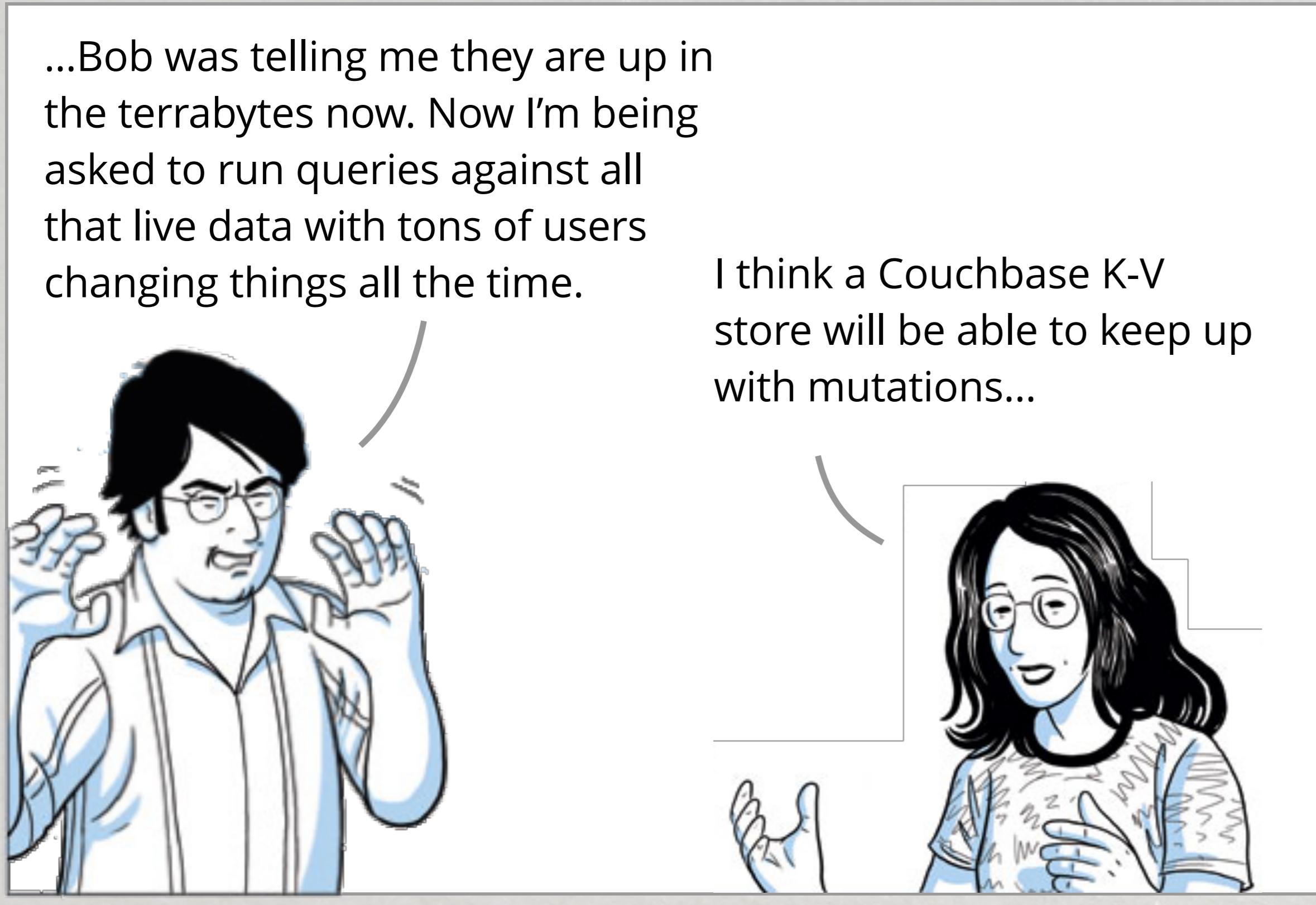
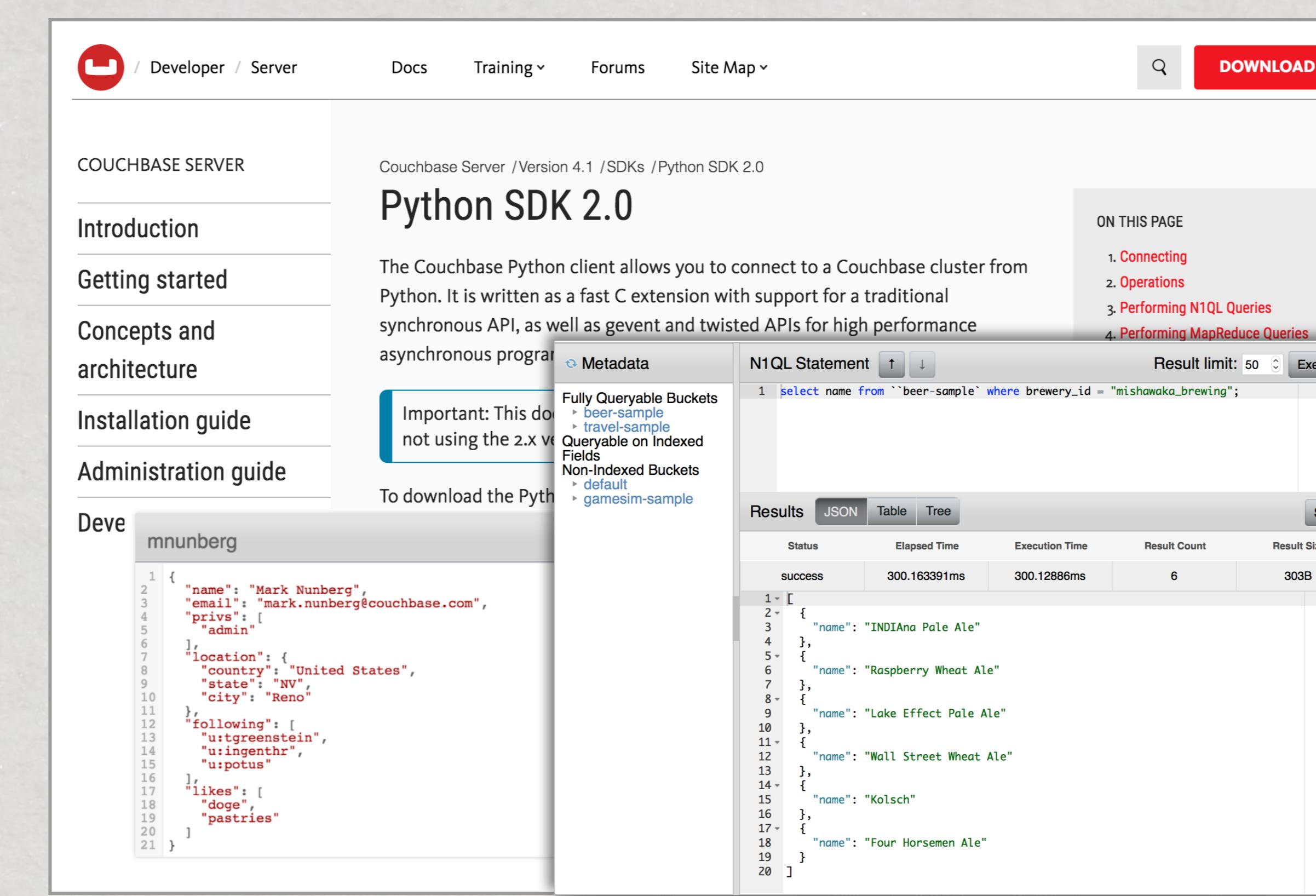


Case 1 - Explore





The Couchbase Python client allows you to connect to a Couchbase cluster from Python. It is written as a fast C extension with support for a traditional synchronous API, as well as gevent and twisted APIs for high performance asynchronous programs.

Important: This document is not using the 2.x documentation. To download the Python SDK 2.0, click here.

ON THIS PAGE

- 1 Connecting
- 2 Operations
- 3 Performing N1QL Queries
- 4 Performing MapReduce Queries

Fully Queryable Buckets

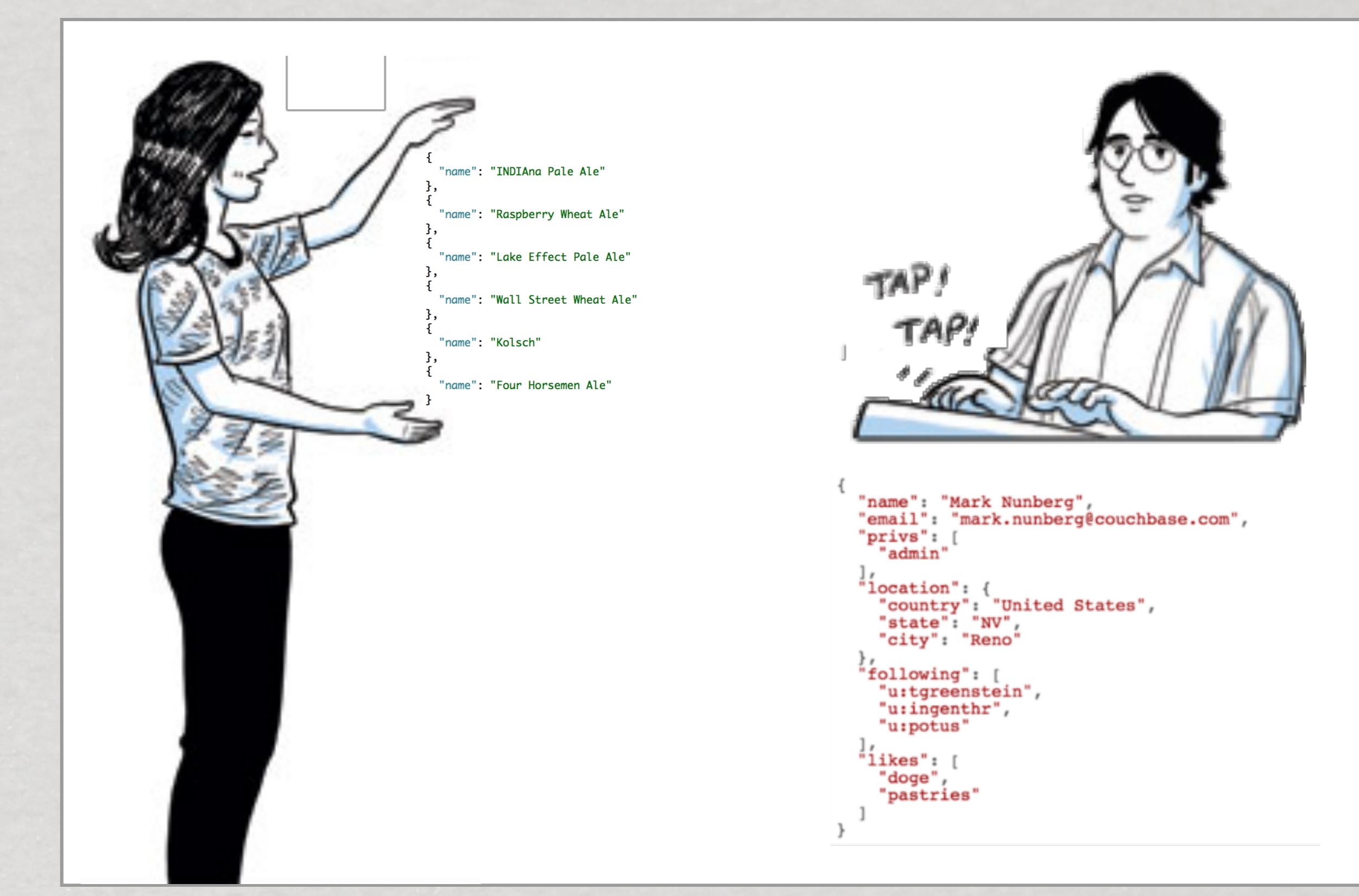
Non-Indexed Buckets

Metadata

Important: This document is not using the 2.x documentation. To download the Python SDK 2.0, click here.

Results

Index	Value	Elapsed Time	Execution Time	Result Count	Result Set
1	success	300.16289ms	300.16289ms	6	2000
2	1				
3	2				
4	3				
5	4				
6	5				
7	6				
8	7				
9	8				
10	9				
11	10				
12	11				
13	12				
14	13				
15	14				
16	15				
17	16				
18	17				
19	18				
20	19				
21	20				



Triggers

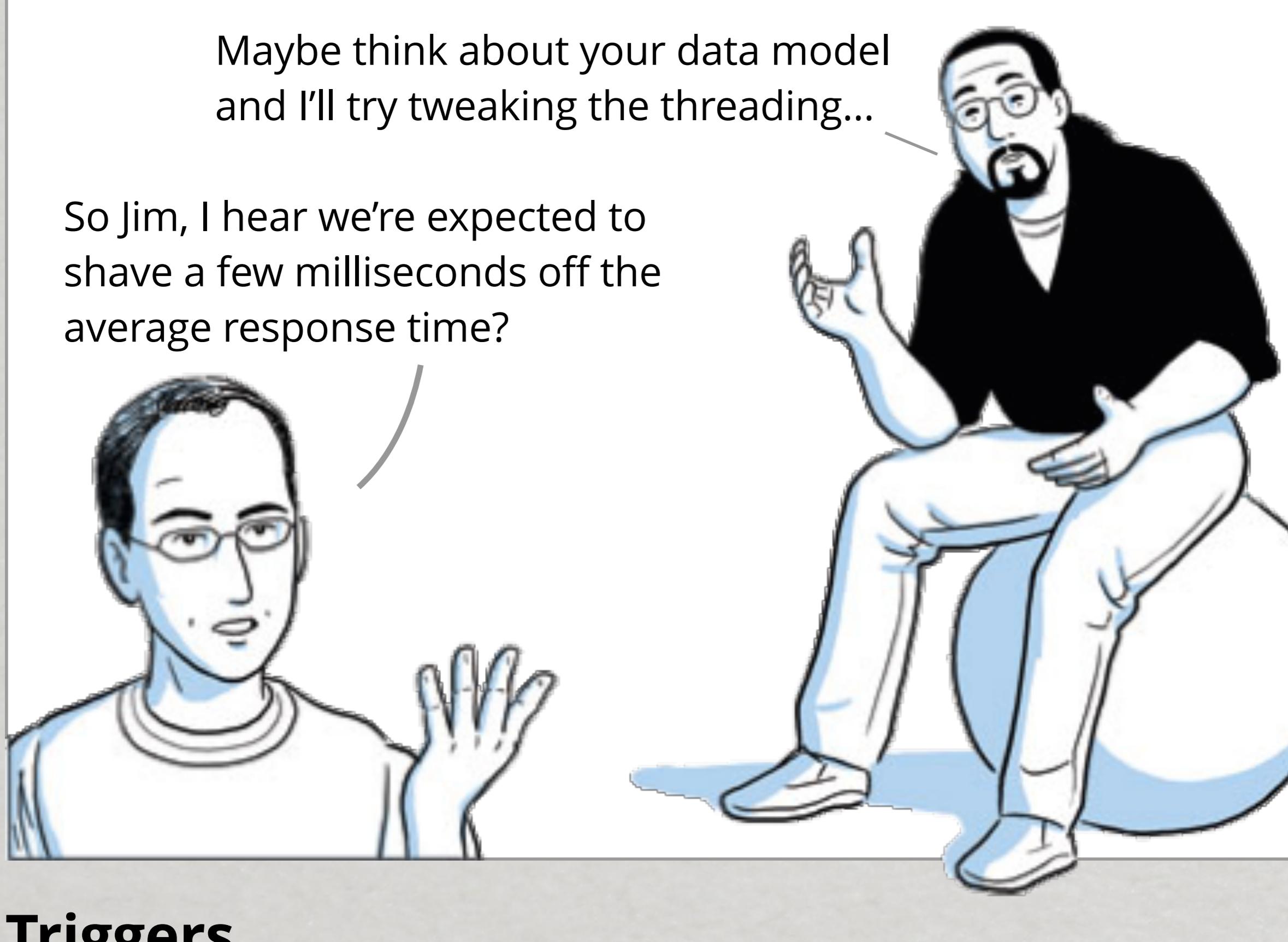
A Developer will explore Couchbase Server because they are evaluating it for some particular purpose, out of sheer curiosity, or—for existing Couchbase users—because they have a new use case for it which requires exploration.

Interfaces
After setup (see Admin Storyboards for setup scenarios), an exploring Developer will make use of Couchbase documentation, sdk/api references, tutorials/sample apps, and couchbase/stack overflow forums. In the web console, they will use sample buckets, Query Workbench, the index pages, and the document viewer/editor.

Frequency & Duration

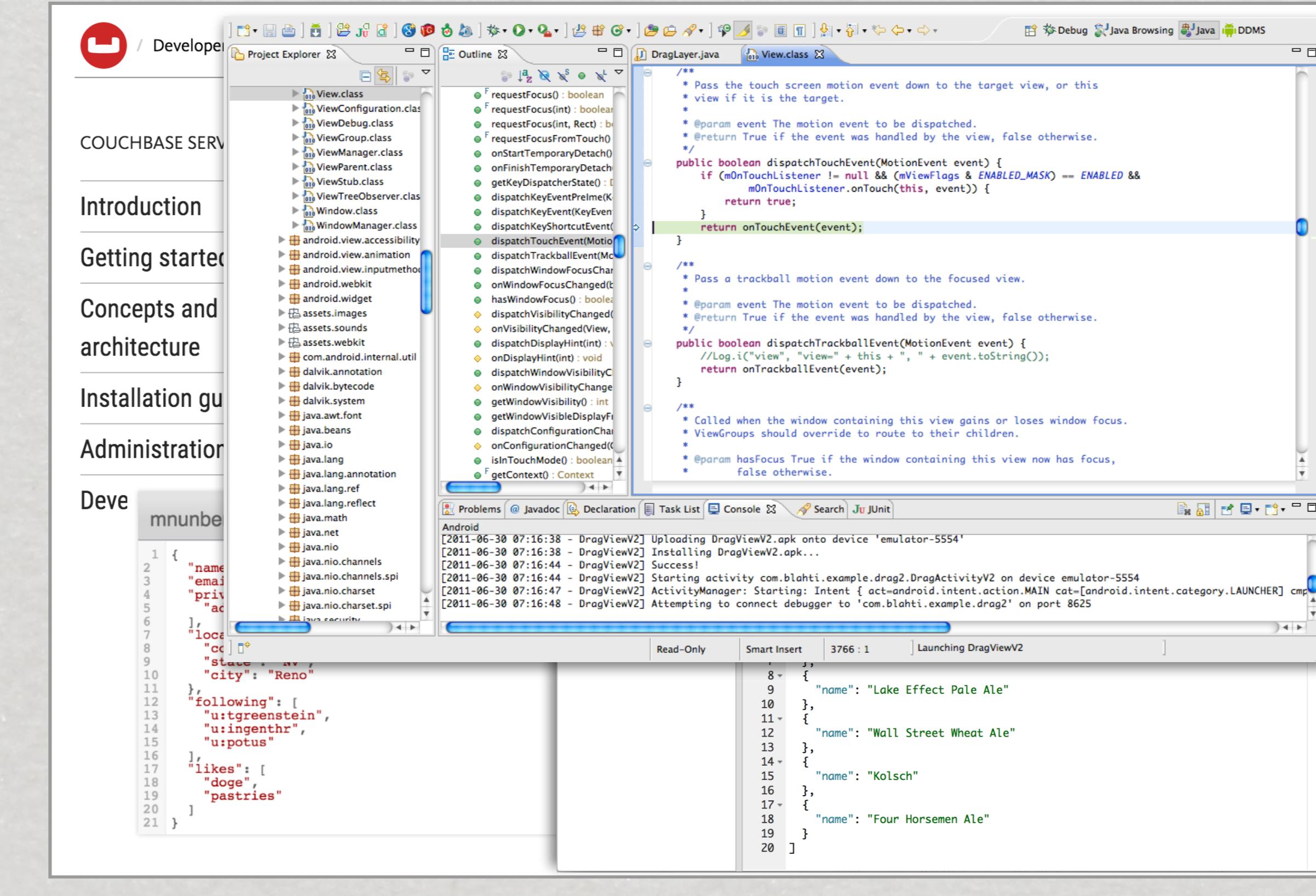
Developer exploration is low frequency and long duration. It is a key phase in developer adoption with many touchpoints.

Case 2 - Optimize

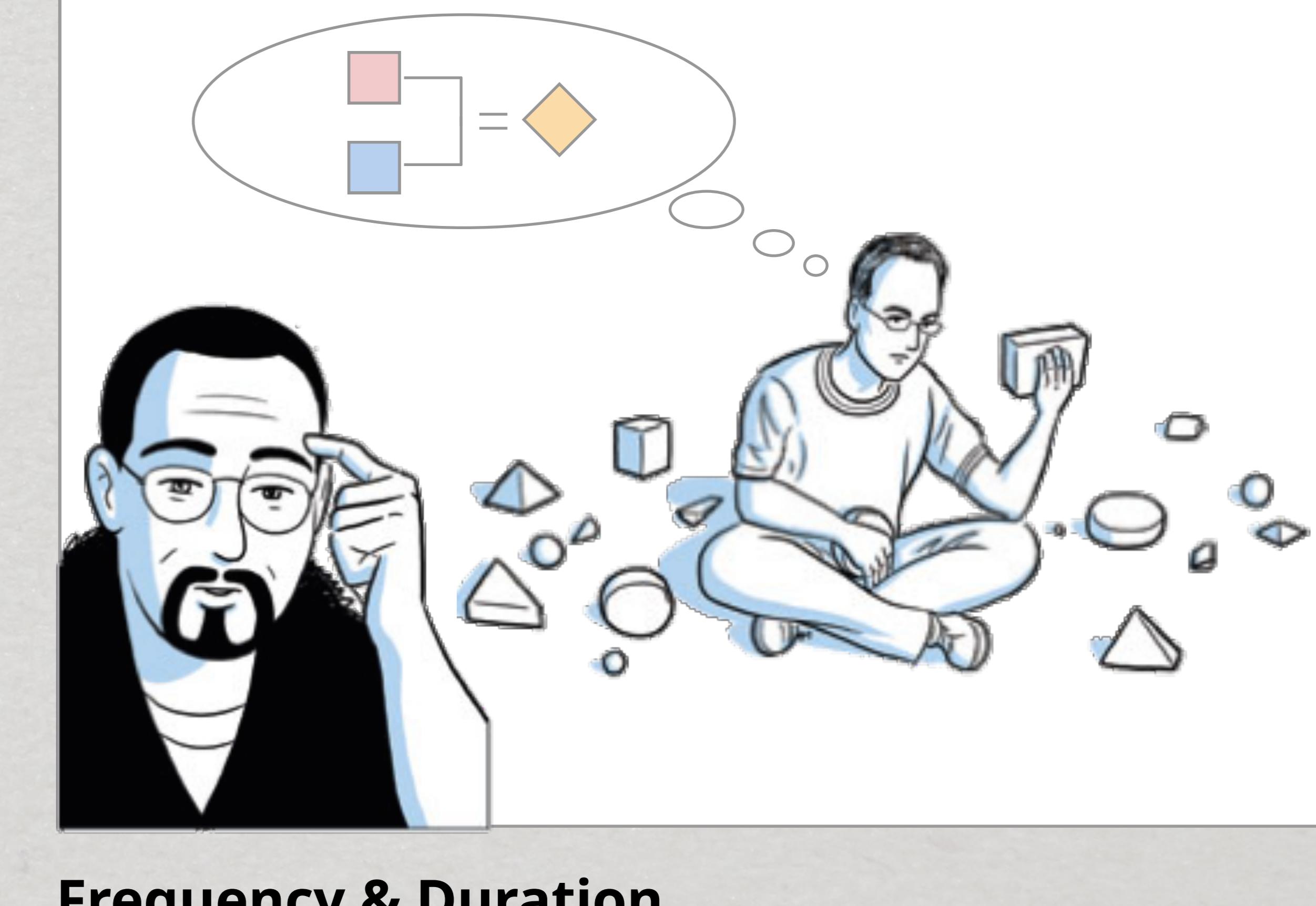


Triggers

A Developer will optimize their app for Couchbase Server when production environments require certain performance guarantees and/or developers are expected to maximize the efficiencies of existing server/cluster capacity.



The code editor shows Java code for a mediator. The logs window displays numerous entries related to mediator tasks and events.



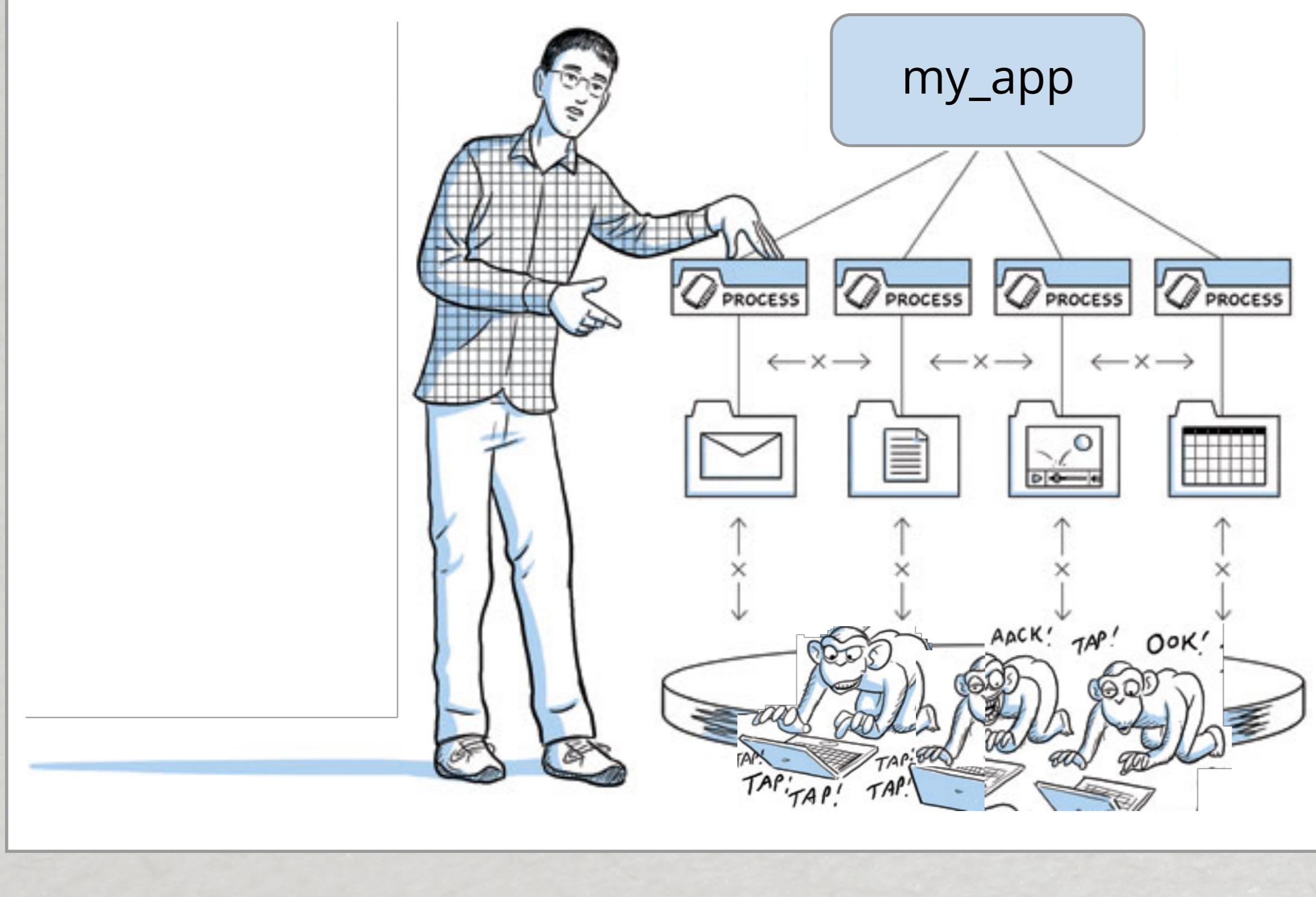
Interfaces

The Developer will optimize their code using their preferred IDE and Couchbase SDK, Couchbase documentation, the Query Workbench, the bucket and server stats pages, and the document viewer/editor.

Frequency & Duration

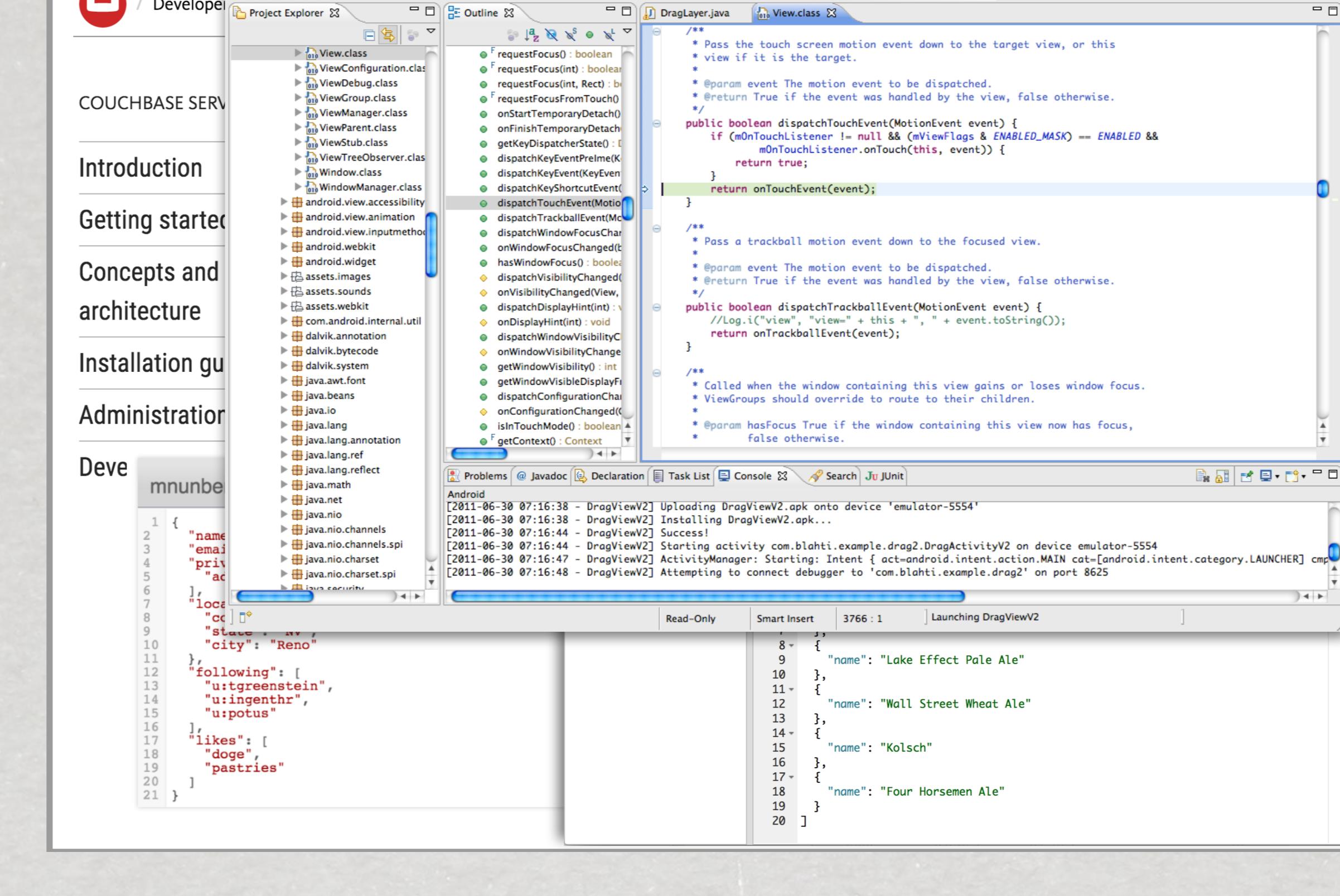
Optimizing is medium frequency and medium duration—assuming a fairly positive outcome. See Analyze & Fix for a more challenging scenario.

Case 3 - Analyze & Fix



Triggers

A Developer is asked to analyze and fix a problem when something fails, the system slows down, becomes corrupted, or any other mysterious condition which cause the application to run poorly or not at all indicates something is wrong with the application code (rather than with Couchbase Server itself).



The code editor shows Java code for a mediator. The logs window displays numerous entries related to mediator tasks and events.



Interfaces

The Developer will analyze and fix their code using their preferred IDE and Couchbase SDK, Couchbase documentation, the Query Workbench, the bucket and server stats pages, and the document viewer/editor.

Frequency & Duration

Analyze & Fix is an emergency situation which should happen VERY infrequently. Finding the fix should not take long, but occasionally mysterious, asymptotic problems will happen which take longer.