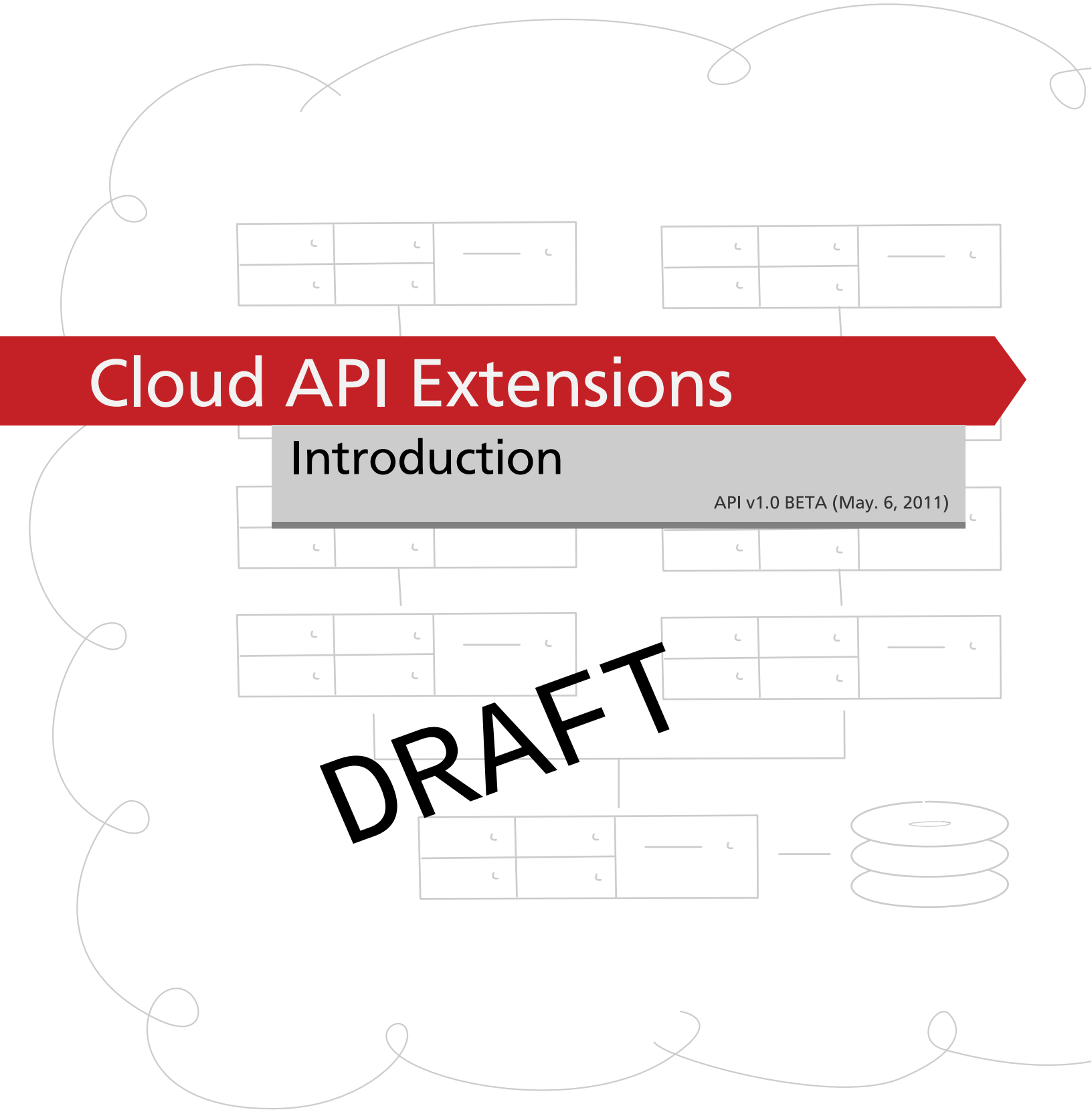
Cloud API Extensions

Introduction

API v1.0 BETA (May. 6, 2011)



rackspace HOSTING

docs.rackspacecloud.com/api

Cloud API Extensions Introduction

API v1.0 BETA (2011-05-06)

This document is intended for software developers interested in developing applications using OpenStack API Extensions The document is for informational purposes only and is provided "AS IS."

Table of Contents

List of Figures

1. Overview

Rackspace Cloud API Extensions make it possible to add functionality to Cloud APIs without modifying the core of the API itself.

We welcome feedback, comments, and bug reports at support@rackspacecloud.com.

1.1. Intended Audience

This Guide is intended to assist software developers who want to develop applications using the Cloud API Extension. To use the information provided here, you should first have a general understanding of Rackspace Cloud APIs. You should also be familiar with:

- ReSTful web services
- HTTP/1.1 conventions
- JSON and/or XML data serialization formats
- Atom syndication format

1.2. Document Change History

This version of the Introduction replaces and obsoletes all previous versions. The most recent changes are described in the table below:

Revision Date	Summary of Changes
May 6, 2011	• Initial draft.

1.3. Additional Resources

You can read the most current versions of other API-related documents at http://docs.rackspacecloud.com/api/.

2. What Problems Can API Extensions Solve?

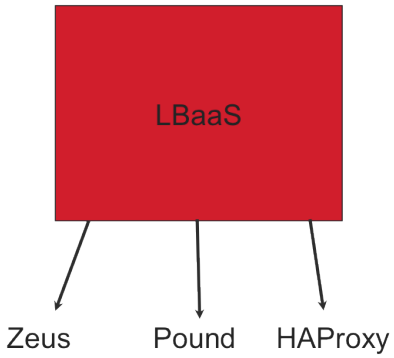
2.1. Standardization vs Innovation and Differentiation

placeholder for content

2.2. Pluggability

placeholder for content

Figure 2.1. One service can reach many backends.



2.3. Open Source

placeholder for content

DRAFT - DRAFT - DRAFT -

3. What Are API Extensions?

3.1. Case Study: Open GL

placeholder for content

3.2. Extensions

placeholder for content

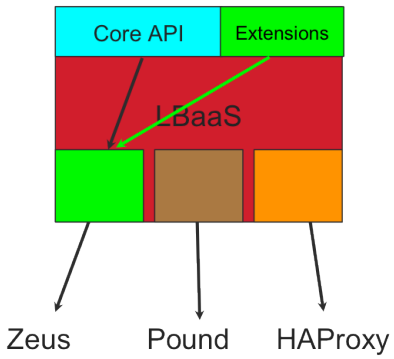
3.3. Extensions vs Versions

placeholder for content

3.4. Extensions and Pluggability

placeholder for content

Figure 3.1. One backend can be reached by both the Core API and Extensions.



DRAFT - DRAFT - DRAFT -

4. How Do API Extensions Work with ReST?

4.1. What can be extended?

placeholder for content

4.2. How are available extensions identified?

placeholder for content

4.3. How is an extension's vendor identified?

placeholder for content

Figure 4.1. Two extensions add a "shared" attribute; one overrides the other.

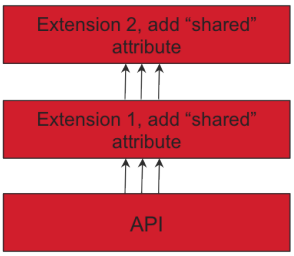
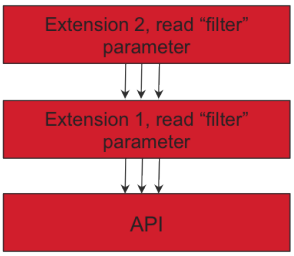


Figure 4.2. Two extensions read a "filter" parameter; both are activated.



4.4. Can data be extended?

placeholder for content

4.5. Can actions be extended?

placeholder for content

4.6. Can headers and states be extended?

placeholder for content

DRAFT - DRAFT - DRAFT -

4.7. Which resources can use which extensions?

placeholder for content

DRAFT - DRAFT - DRAFT -

5. How Can an API Extension Become an API Feature?

5.1. Specification governance

placeholder for content

5.2. ARB-Approved Extensions

placeholder for content

5.3. Promotion Path

placeholder for content

DRAFT - DRAFT - DRAFT -

6. What Challenges Must Be Met?

6.1. Language Bindings

placeholder for content

6.2. JSON and Collections

placeholder for content

DRAFT - DRAFT - DRAFT -

7. When is an API Extension the Best Approach?

7.1. Why use extensions?

placeholder for content

7.2. How to use extensions?

placeholder for content

DRAFT - DRAFT - DRAFT -