

Upgrade Your APEX App with Zero Downtime by Using EBR

Bryn Llewellyn

Distinguished Product Manager

Database Division

Oracle HQ

twitter: @BrynLite



Joel Kallman

Senior Director of APEX Development

Database Division

Oracle USA

twitter: @joelkallman

Spring 2017

ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

APEX Connect – Spring 2017

APEX, together with the ORDS server, lets you build applications: (a) with slick browser interfaces that are deployed on smartphones and the like; and (b) access other RESTfully exposed services. Moreover, the Oracle “PDBaaS” Cloud Service provisions and manages all the infrastructure for you so all you have to do is pure development.

You patch and upgrade such an app with zero downtime by deploying a new version of your APEX app alongside the running version, configuring it to use a new edition in which you make your ordinary database changes. When everything is ready, you use the ORDS server to direct sessions that start after this moment to the new version; meanwhile, established sessions automatically continue to use the old version until they end naturally.

Let us, Bryn and Joel, tell you how to do this.

Agenda

- 1 Edition-Based Redefinition (EBR)
- 2 APEX & EBR together
- 3 Demos
- 4 Conclusion

Agenda

- 1 Edition-Based Redefinition (EBR)
- 2 APEX & EBR together
- 3 Demos
- 4 Conclusion

Brief summary of EBR

- EBR lets you patch or upgrade the artifacts that implement the persistence for your application, in an Oracle Database, with zero downtime
- It allows the database, from the PoV of client-side code, to look like two databases, each identified in the normal way by a service name
 - just like two PDBs in the same CDB

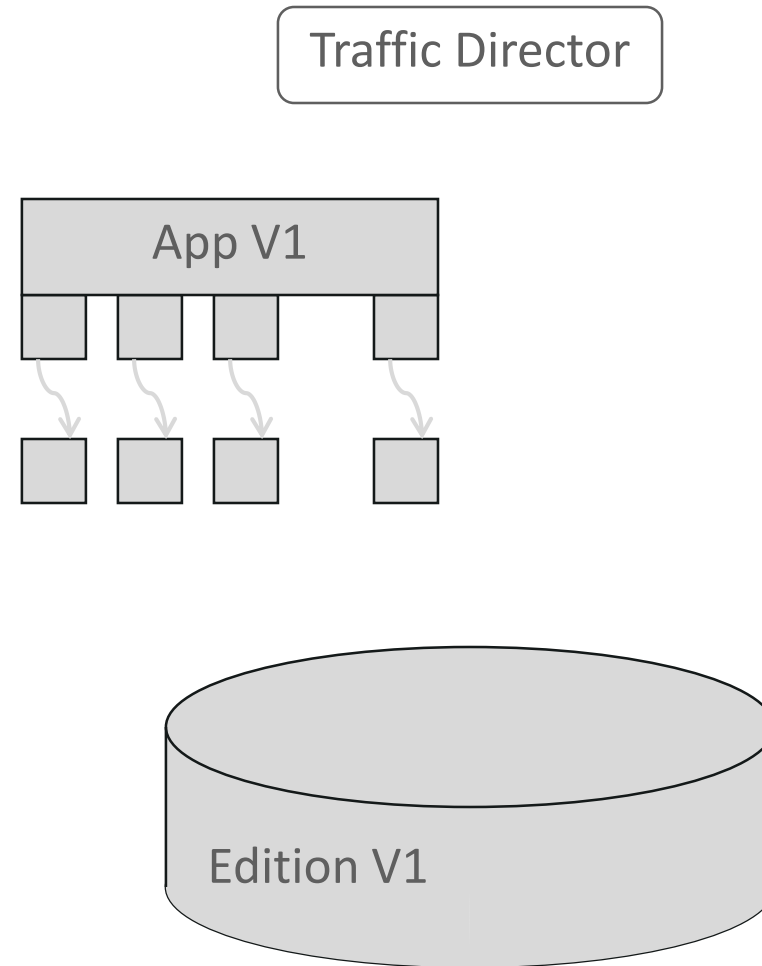
Brief summary of EBR – *cont*

- The new database is instantly created as a new *edition* of the old, using a special sparse copy scheme
- You make your changes in the new edition, while the old edition remains in uninterrupted use
- You change PL/SQL code, views, and synonyms using the same DDLs that you use in offline patching
- Special very efficient mechanisms keep the data in the old and the new editions mutually synchronized, while they're in concurrent use, even when you refactor the way the data is stored, so that you honor data rules in the normal transactional fashion

Brief summary of EBR – *cont*

- When the new edition is ready, you let new end-user sessions start to use it. Meanwhile, established end-user sessions continue to use the old edition until the user is ready to finish
- This way, the whole population of users (human and robotic) rolls over from the old edition to the new so that no session ever experiences an interruption of service

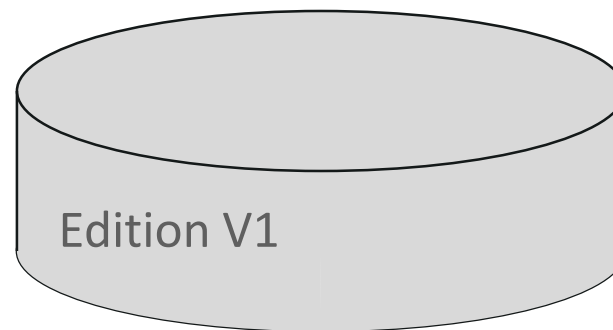
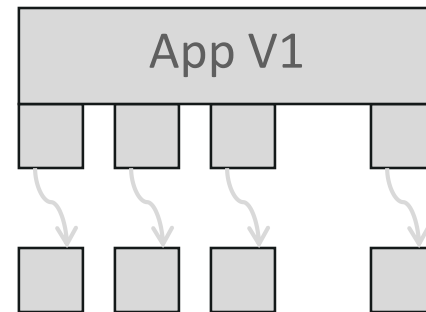
Hot rollover across the stack



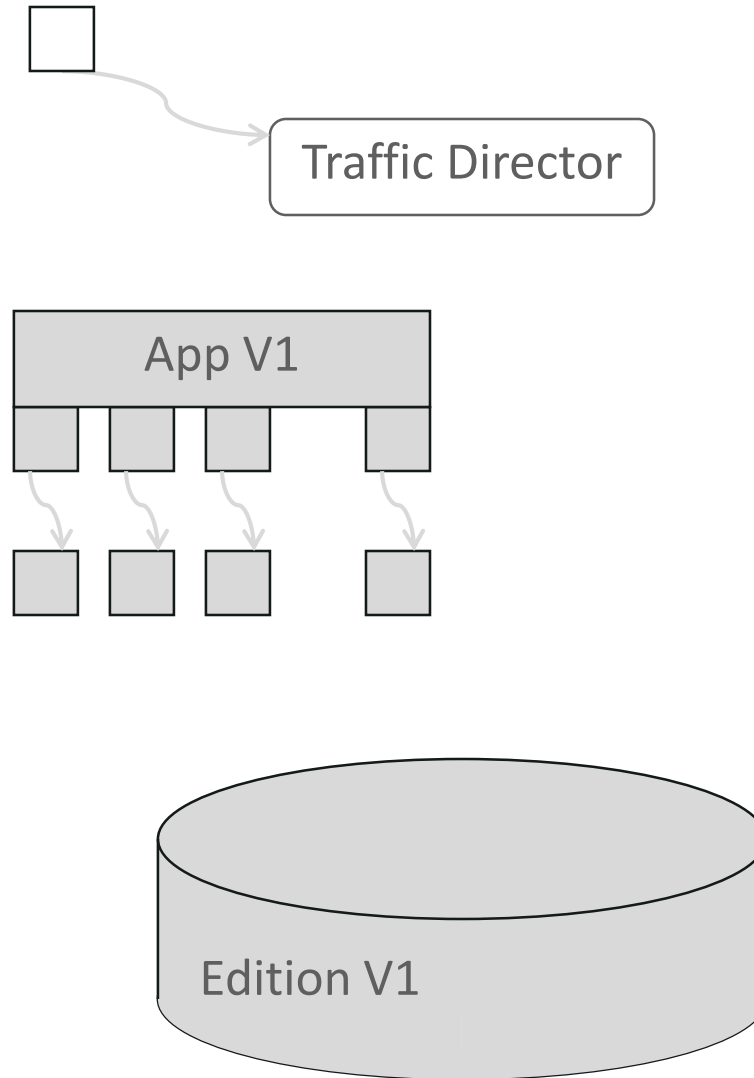
Hot rollover across the stack



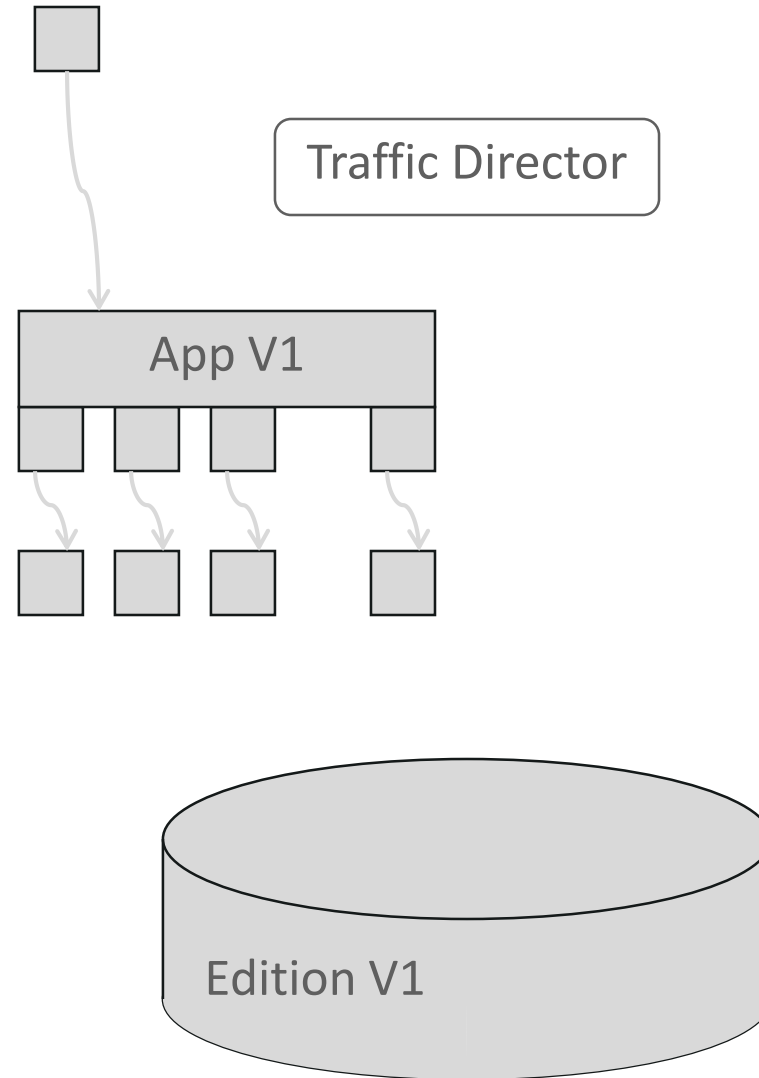
Traffic Director



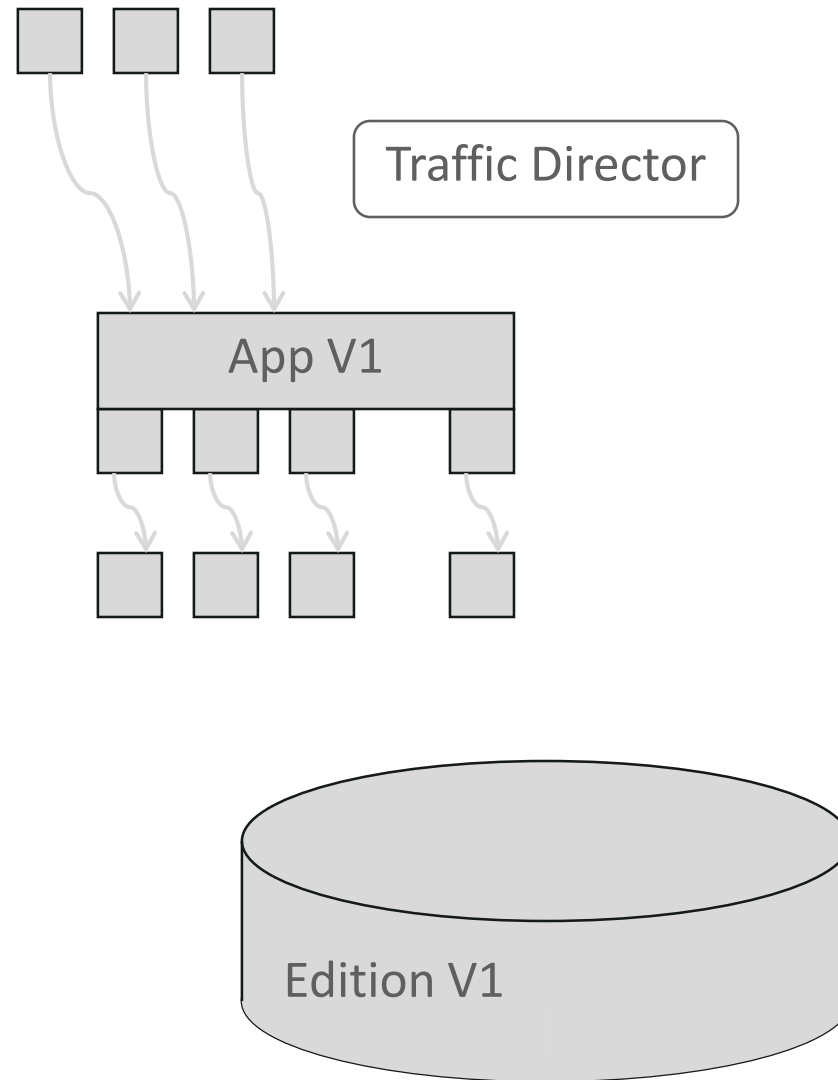
Hot rollover across the stack



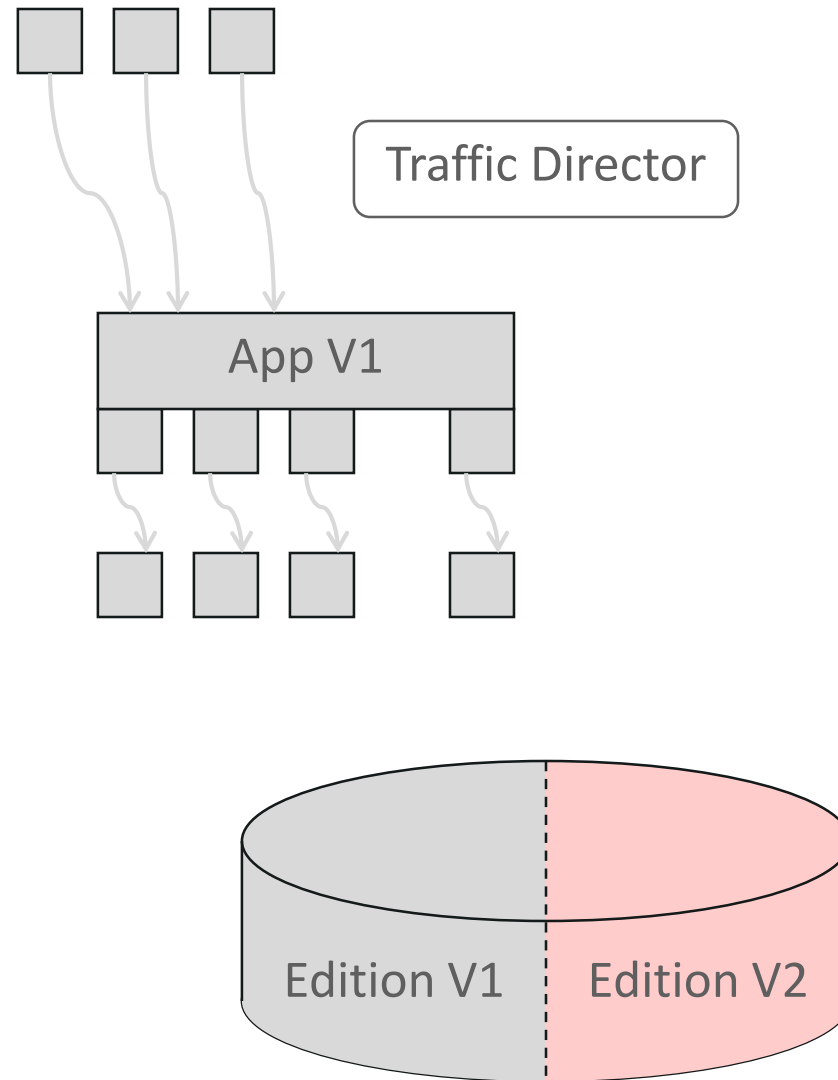
Hot rollover across the stack



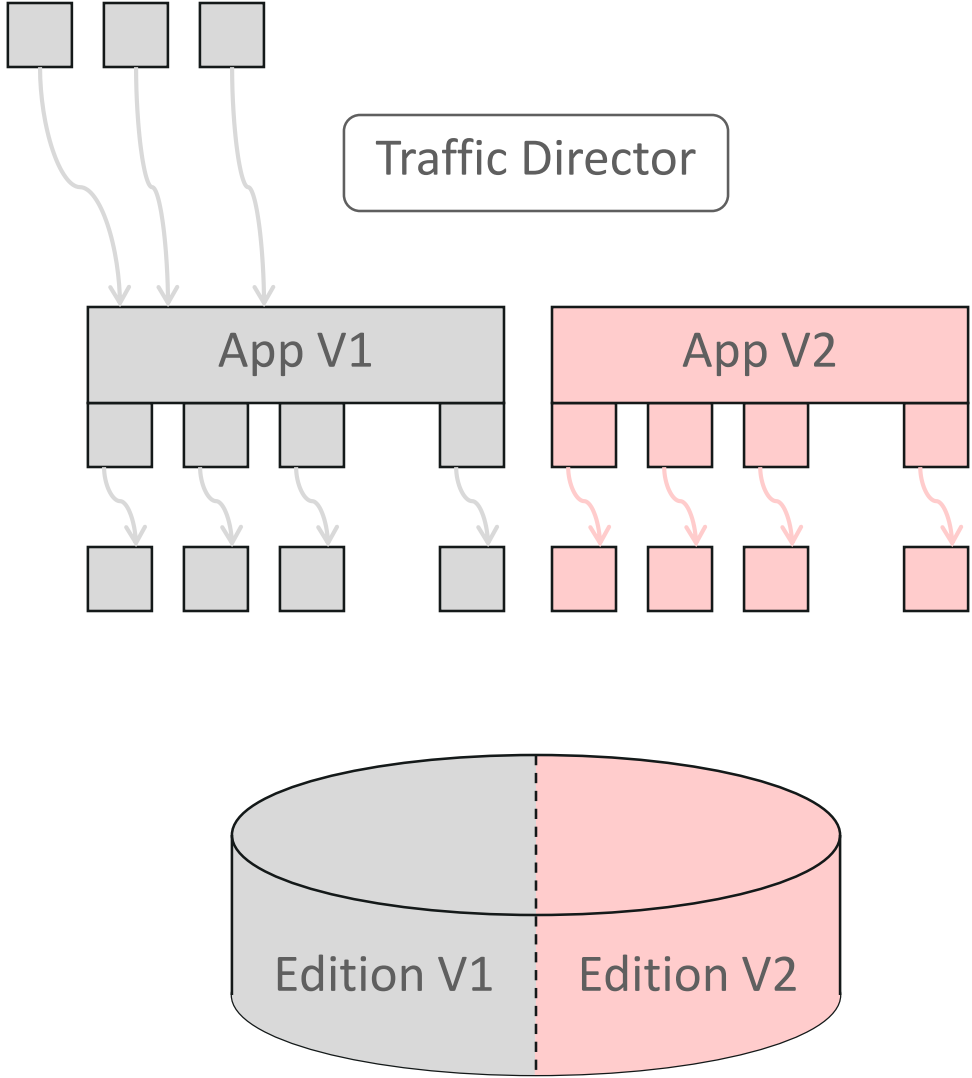
Hot rollover across the stack



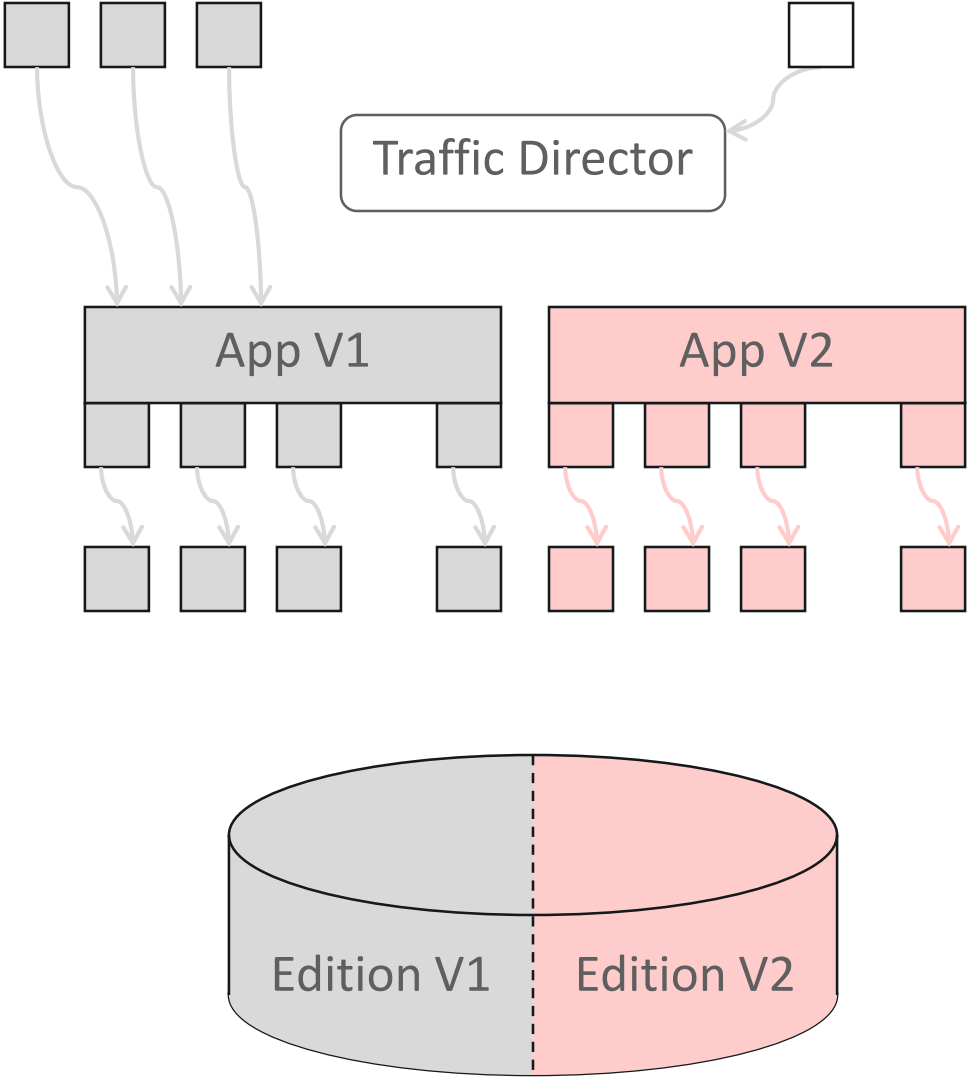
Hot rollover across the stack



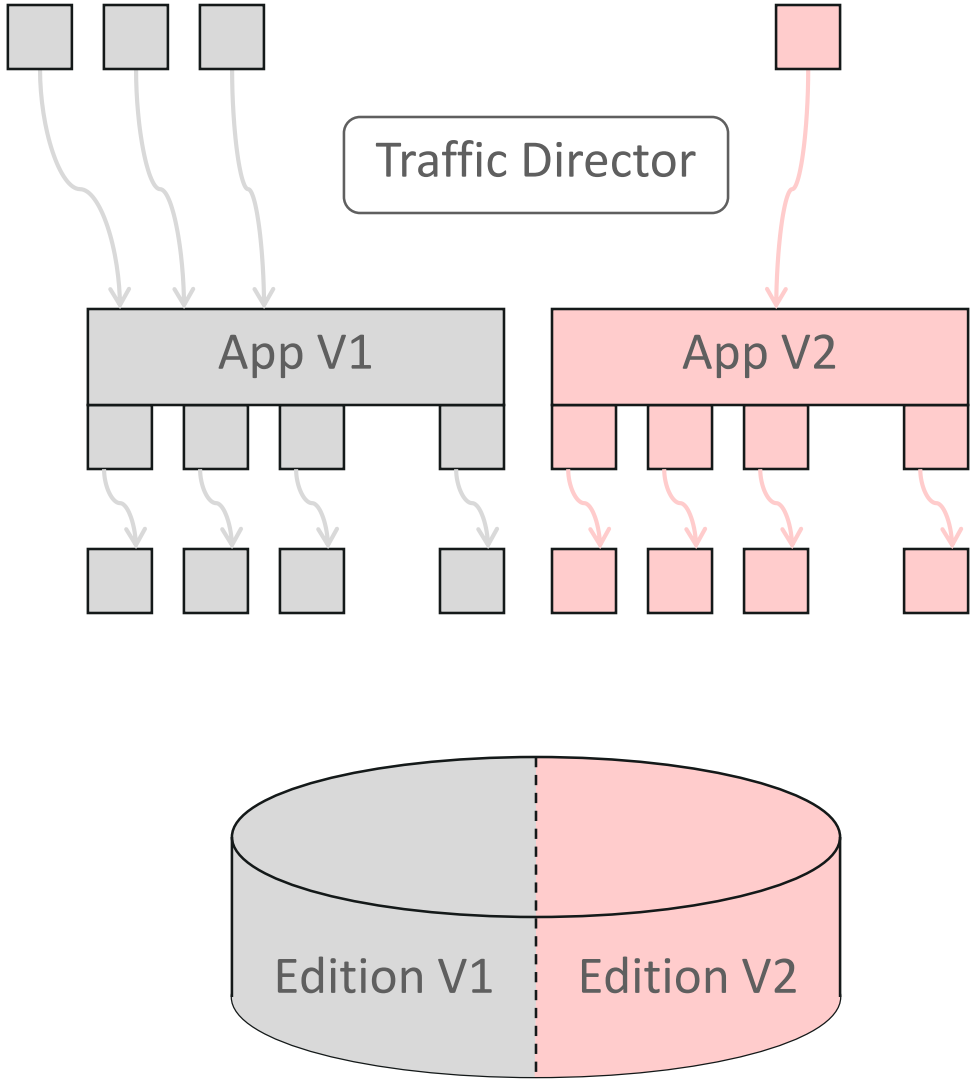
Hot rollover across the stack



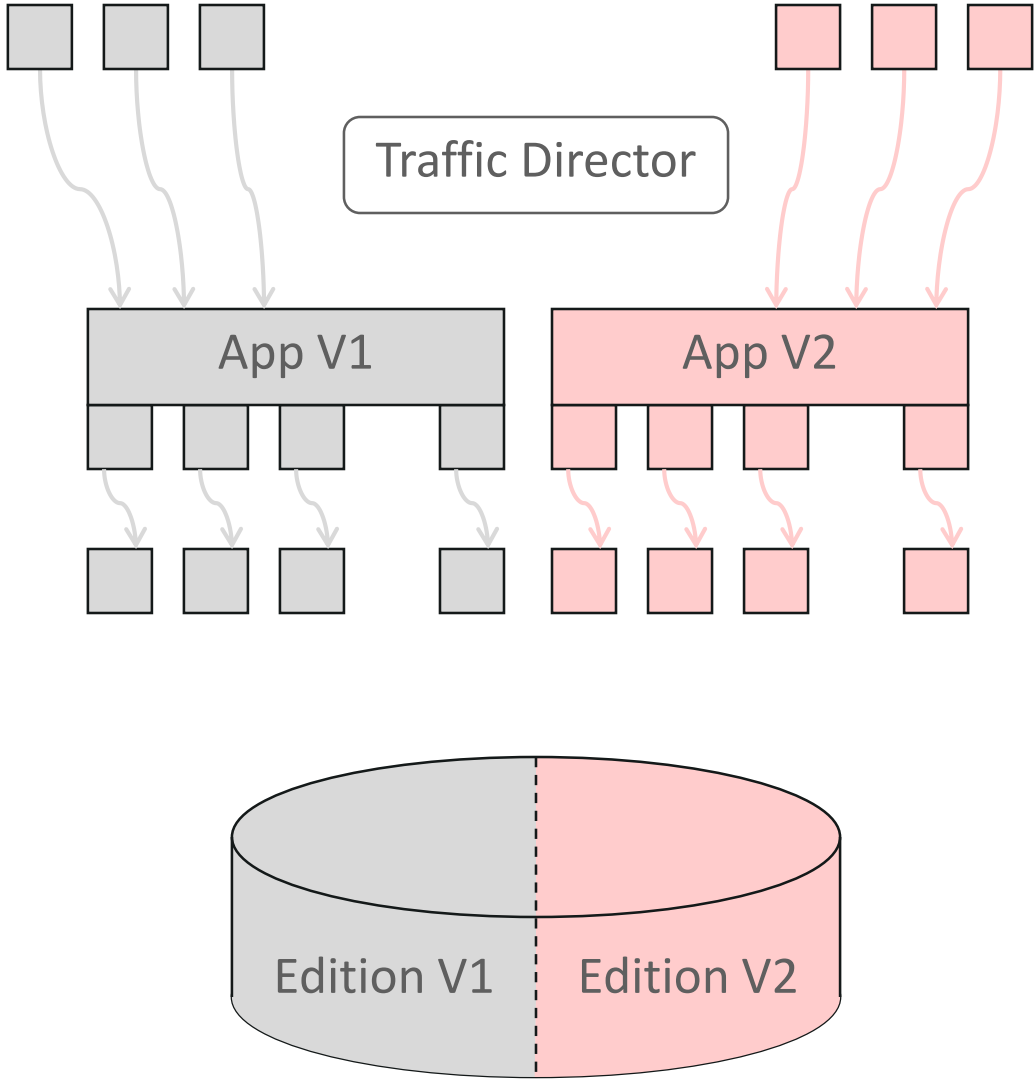
Hot rollover across the stack



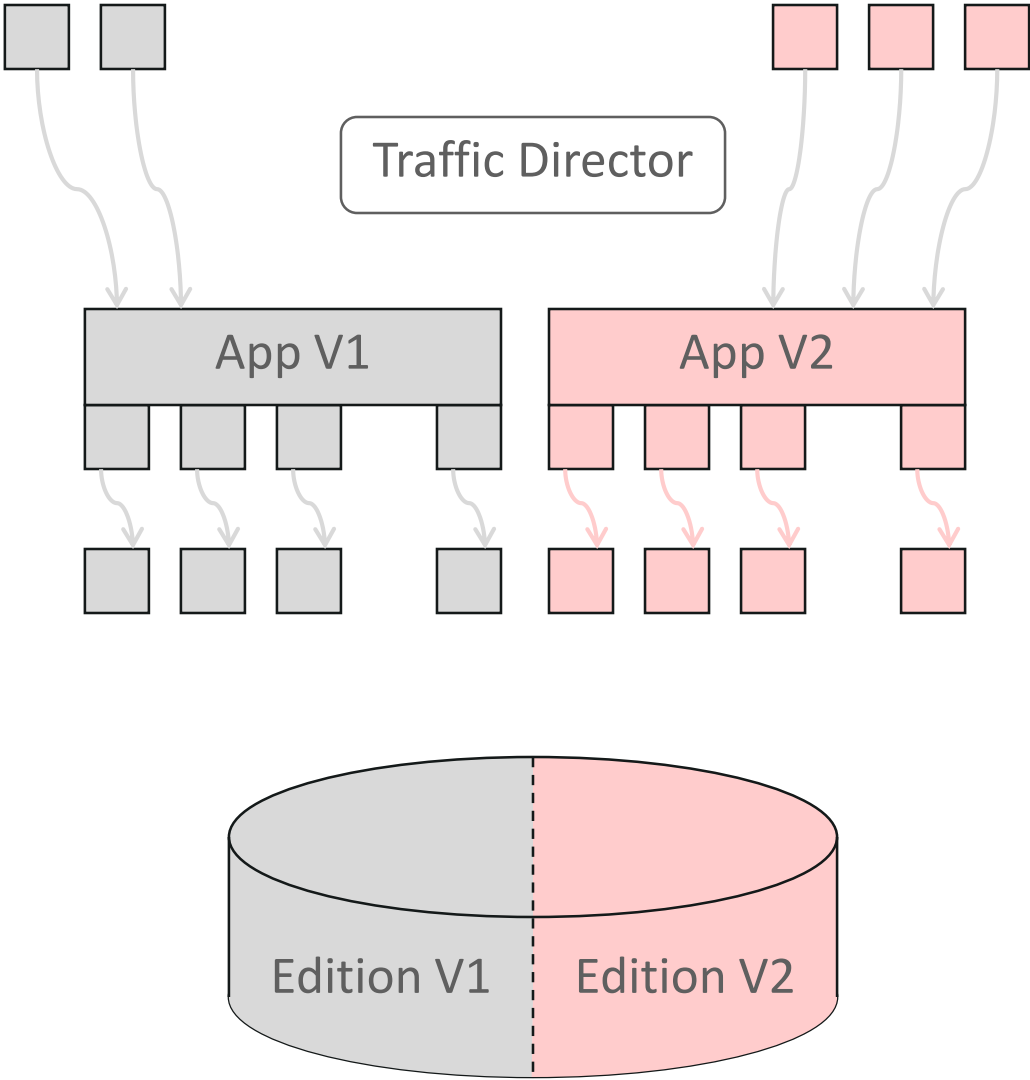
Hot rollover across the stack



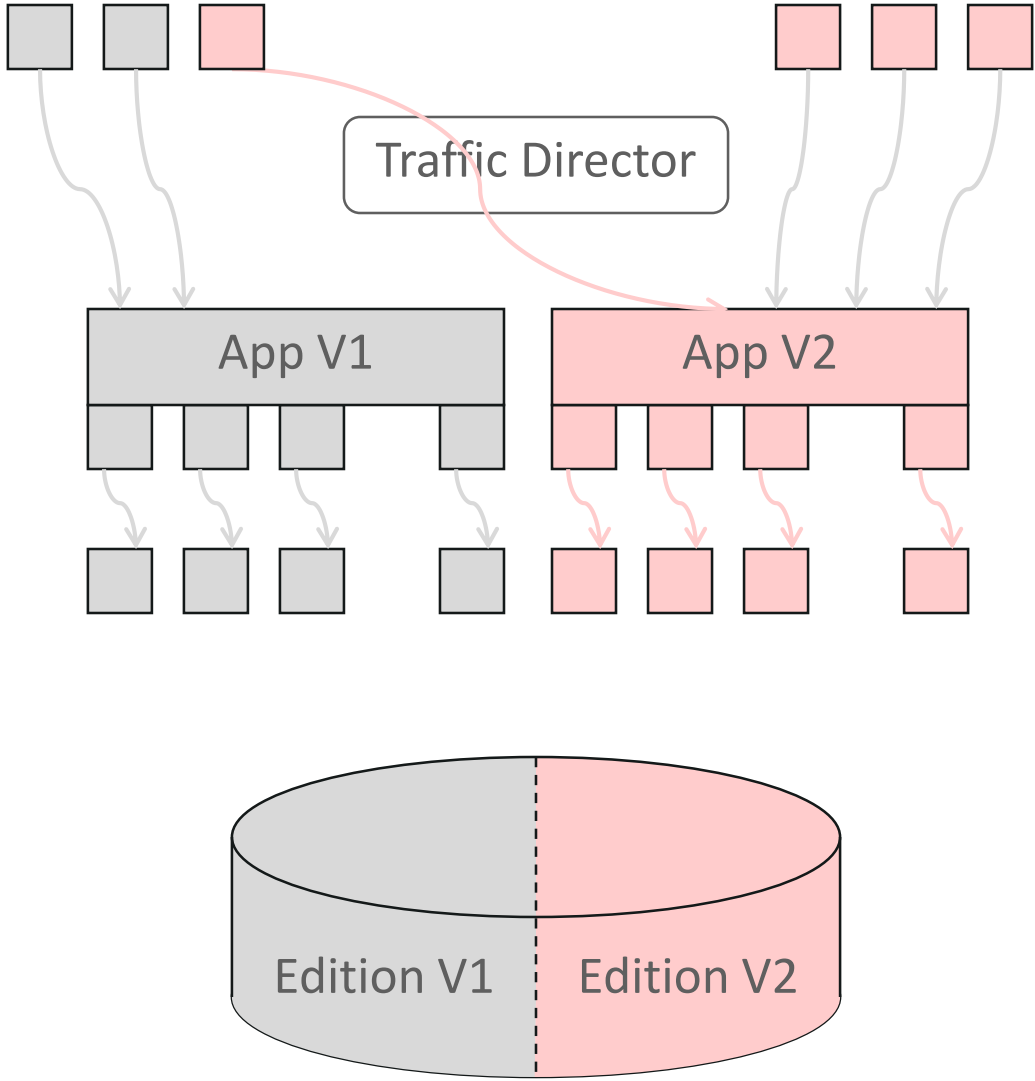
Hot rollover across the stack



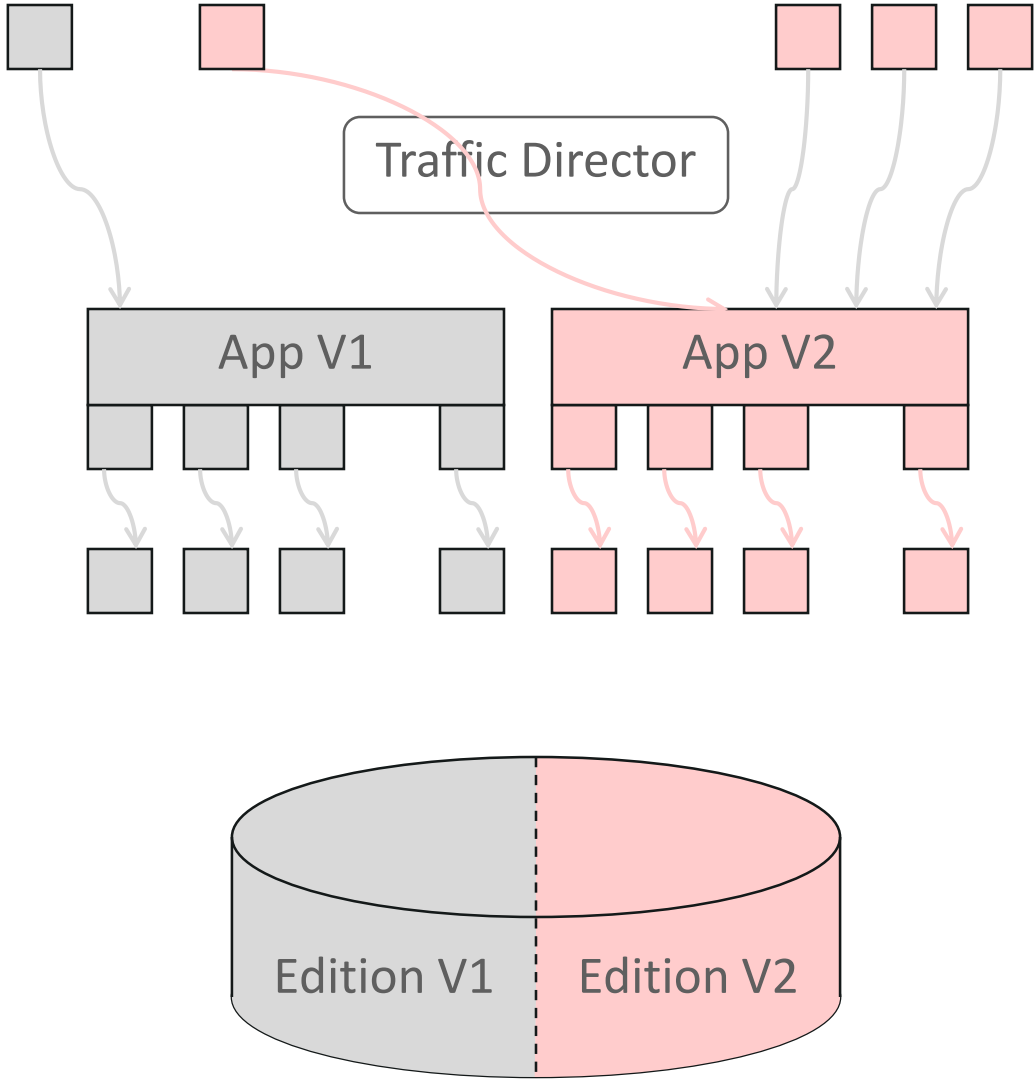
Hot rollover across the stack



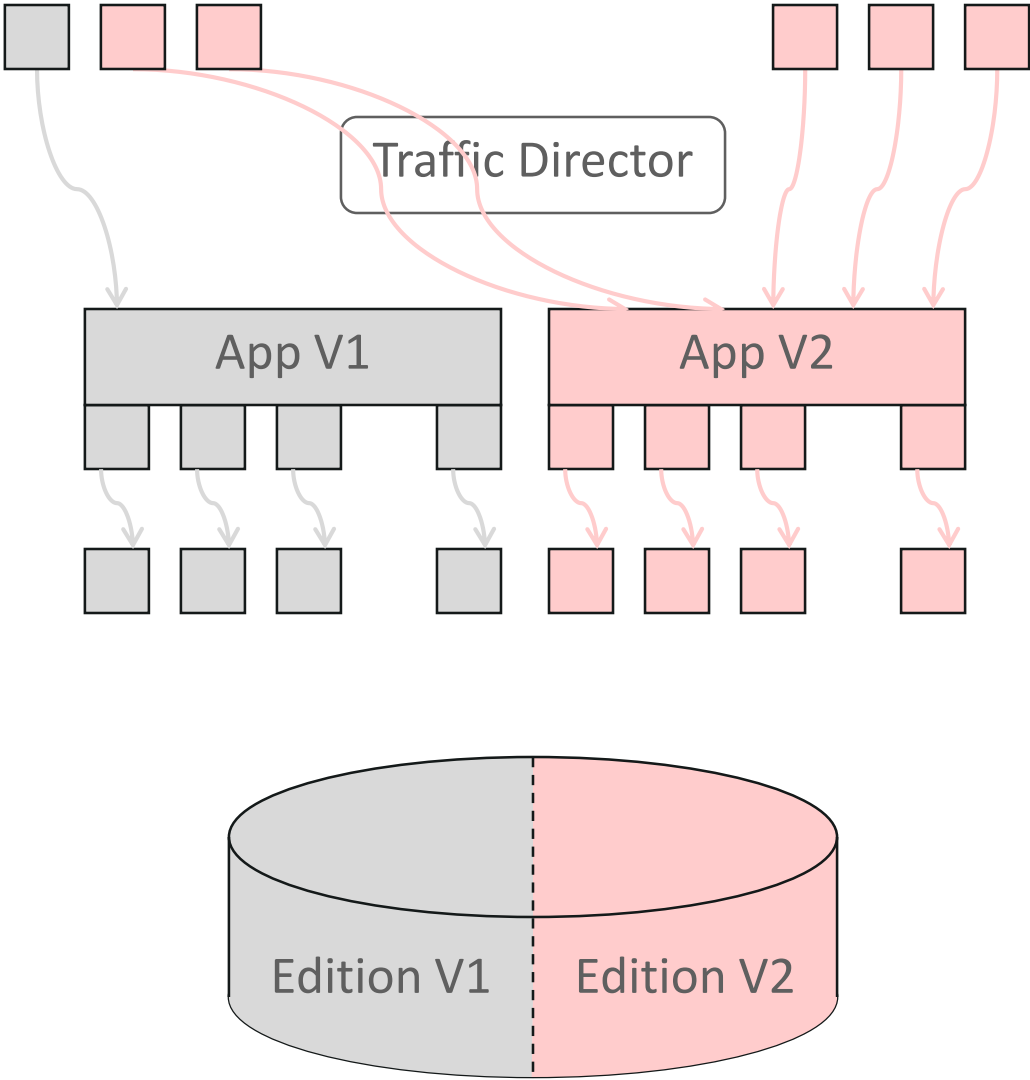
Hot rollover across the stack



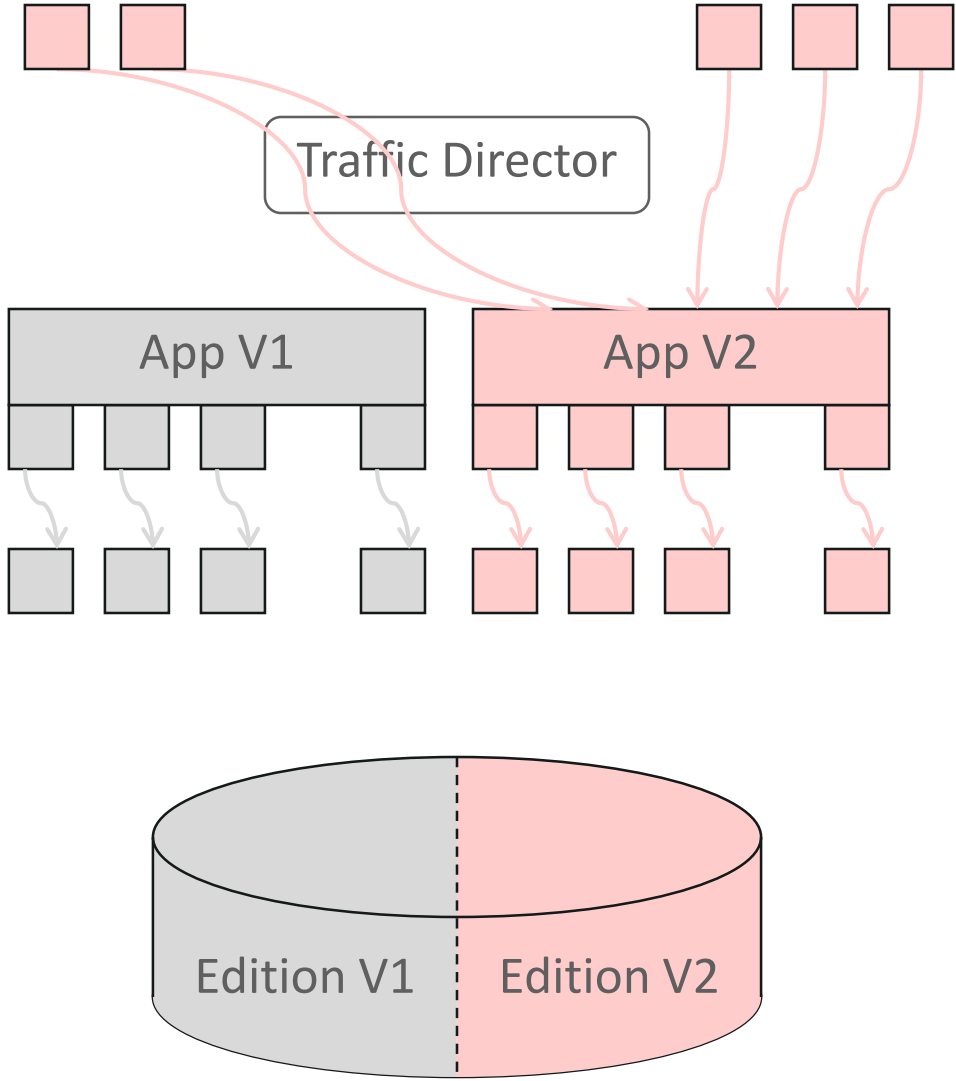
Hot rollover across the stack



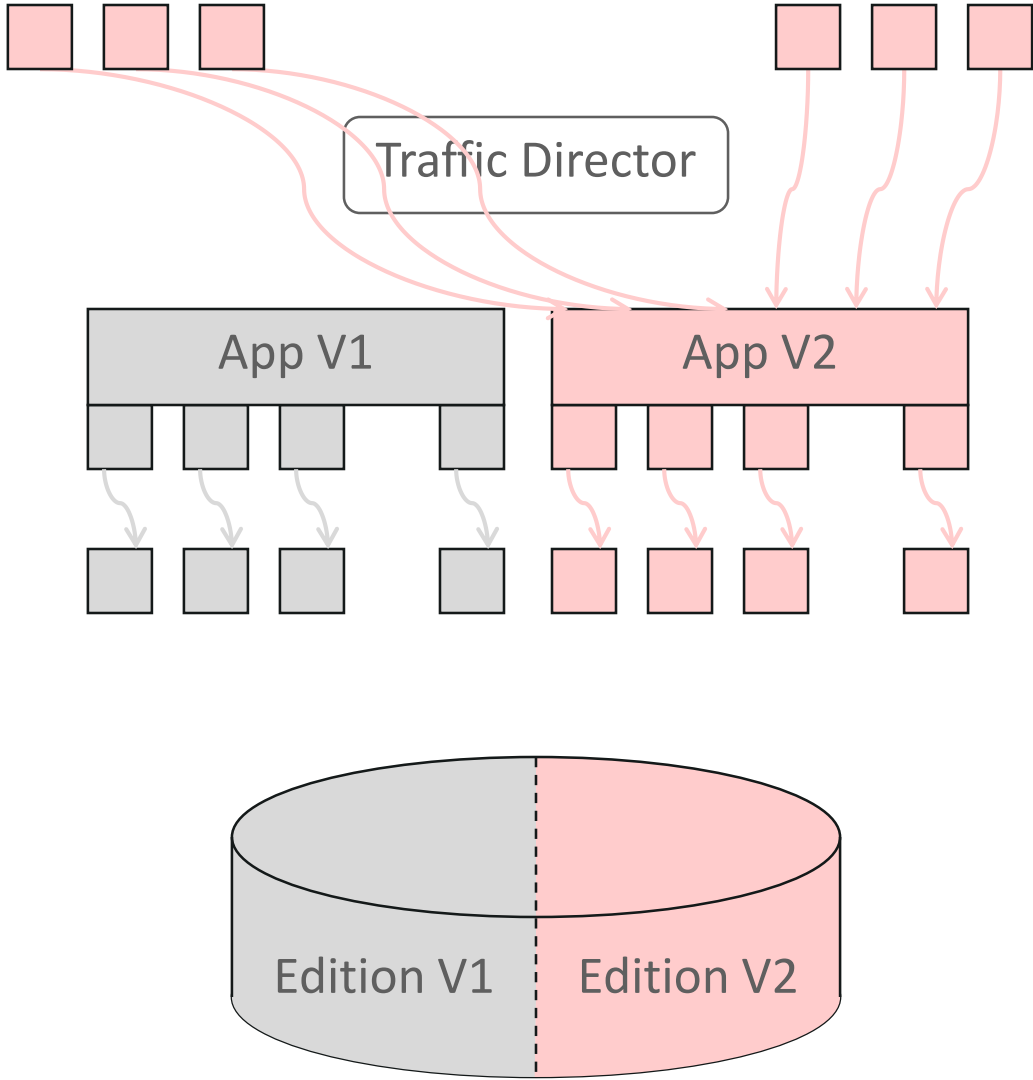
Hot rollover across the stack



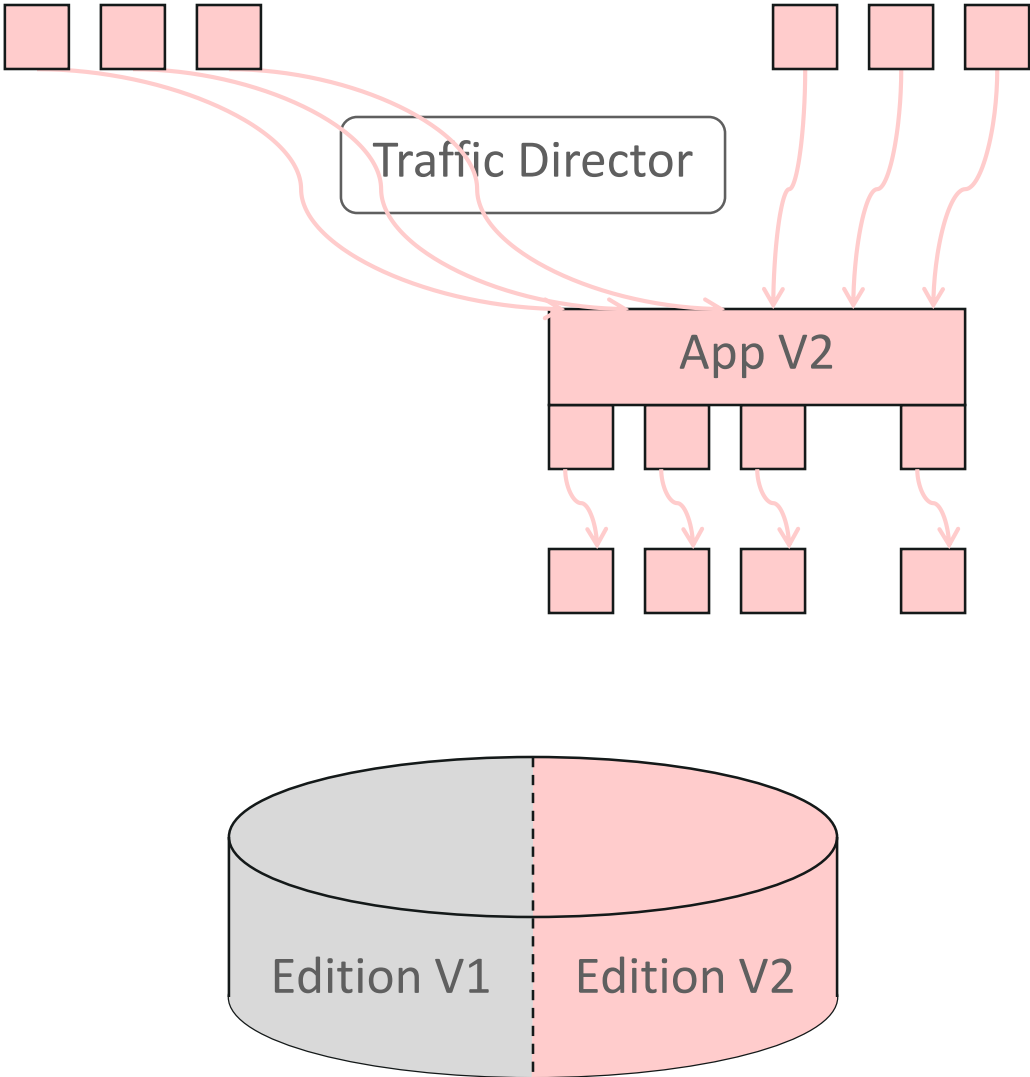
Hot rollover across the stack



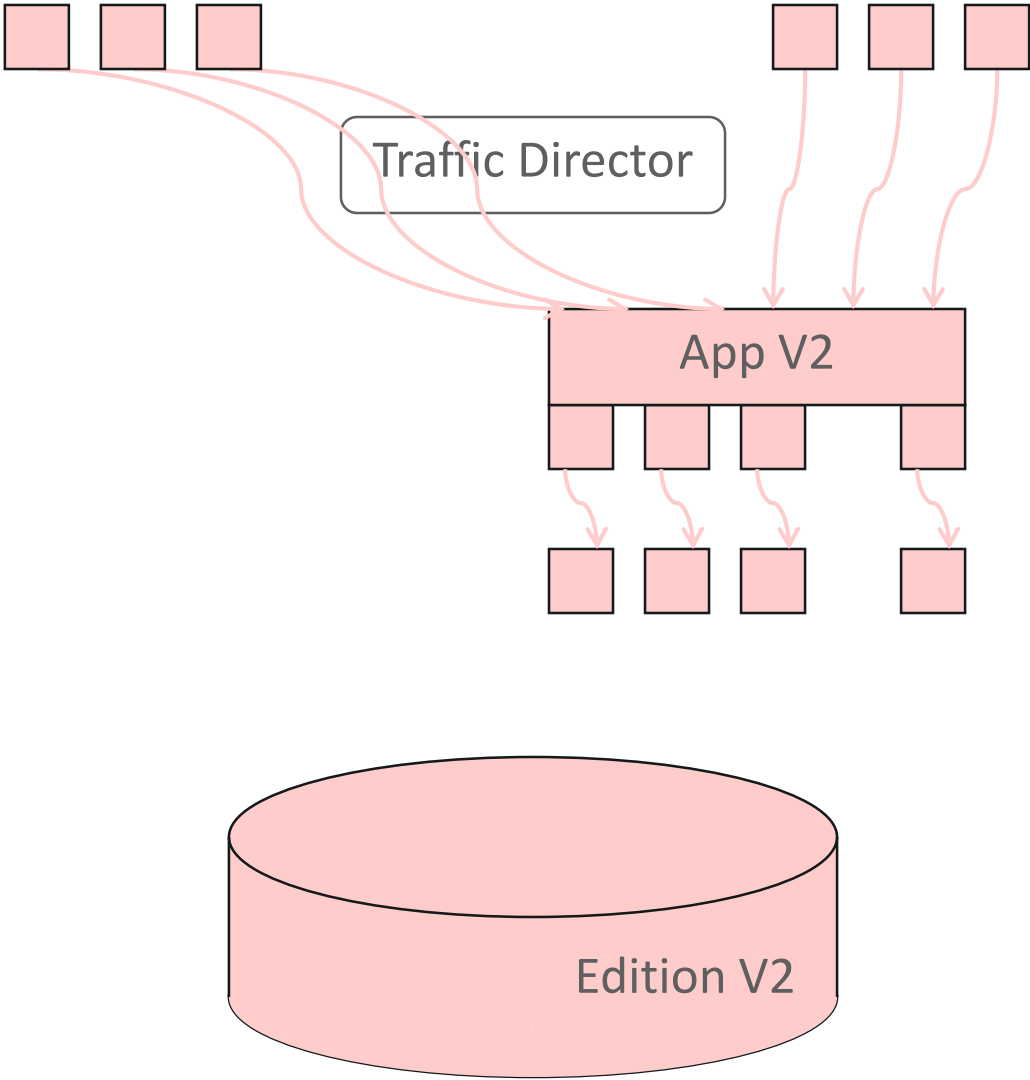
Hot rollover across the stack



Hot rollover across the stack



Hot rollover across the stack



EBR is functionally complete, robust, and performance neutral

- E-Business Suite Release 12.2, GA at OpenWorld 2013, uses EBR to do all patching and upgrading. Several hundred customers are using it in production
- Salesforce.com gave presentations at OpenWorld 2014 and Kscope16 on their adoption of EBR
- AmEx gave a presentation at OpenWorld 2015 on their adoption of EBR
- MoneyGram and Moovit gave presentations at OpenWorld 2016 on their adoptions of EBR
- **If they can do it, so can you!**

Agenda

- 1 Edition-Based Redefinition (EBR)
- 2 APEX & EBR together**
- 3 Demos
- 4 Conclusion

Specializing the general hot rollover model for EBR with APEX

- APEX is already *inside* the database, so the *service* notion isn't available
- Instead, an APEX application specifies the edition it should use as a property
 - Every database call that APEX does uses *DBMS_Sql* so it uses the “*edition trampoline*” overload: *Parse(...Edition=>'e1'...)*
- The APEX dictionary is not editions-enabled
APEX has no explicit versioning notions
so just use a naming convention:
My_App_v1 (using *e1*) -> *My_App_v2* (using *e2*)
- How do we implement *end-user-session* to *application-version* affinity?

End-user-session to application-version affinity for APEX

- Publish a permanent, bookmarkable URL for your APEX app's homepage
- Set up a URL redirection using ORDS to map to the current app version
 - This is done with a simple piece of PL/SQL code
- When the new app version is ready for use, change the redirection to map to this
- The APEX system ensures that every embedded URL in every browser page that it generates encodes the surrogate identifier for the app (rather than its name)
- *End-user-session to application-version* affinity therefore solves itself

Agenda

- 1 Edition-Based Redefinition (EBR)
- 2 APEX & EBR together
- 3 **Demos**
- 4 Conclusion

1. Basic EBR demo

2. APEX demo

Agenda

- 1 Edition-Based Redefinition (EBR)
- 2 APEX & EBR together
- 3 Demos
- 4 **Conclusion**

Conclusion

- You have seen that by using only the skills of an Oracle Database development engineer, you can build a fully functional complete application and patch and upgrade it with zero downtime
- This is especially interesting in Oracle's "PDBaaS" because you get this functionality with a single subscription and without having to be concerned about the mundane administration tasks

General PL/SQL and EBR resources

- PL/SQL page on OTN: `oracle.com/plsql`
- EBR page on OTN: `oracle.com/ebr`
- Oracle Database PL/SQL and EBR blog: `blogs.oracle.com/plsql-and-ebr/`

The Oracle logo consists of the word "ORACLE" in white, uppercase, sans-serif font, centered within a solid red rectangular background.

ORACLE®

Integrated Cloud
Applications & Platform Services