



MERGE MADNESS

Dimitri Gielis & Jackie McIlroy
ODTUG Kscope24
Nashville, TN

Extend APEX like a PRO



APEX Office Print

Printing & Reporting
built for APEX



APEX Media Extension

Image processing for the
Oracle Database.



APEX Office Edit

Document management
for APEX



Plug-ins Pro

Document management
for APEX

APEX Message Service

Real-time communication
for Oracle APEX



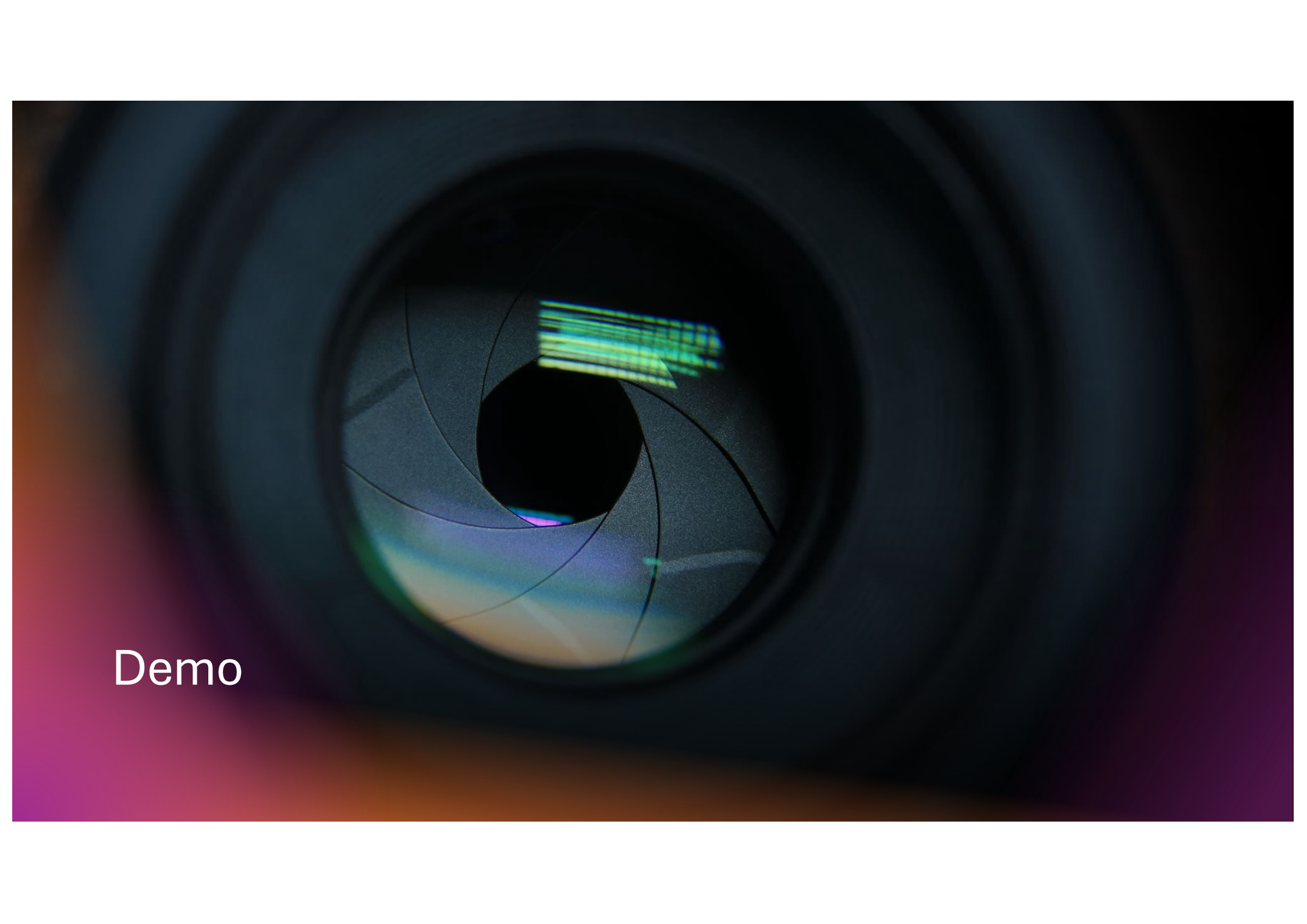
APEX Project Eye

Productivity, Quality Assurance,
and Security tool for APEX

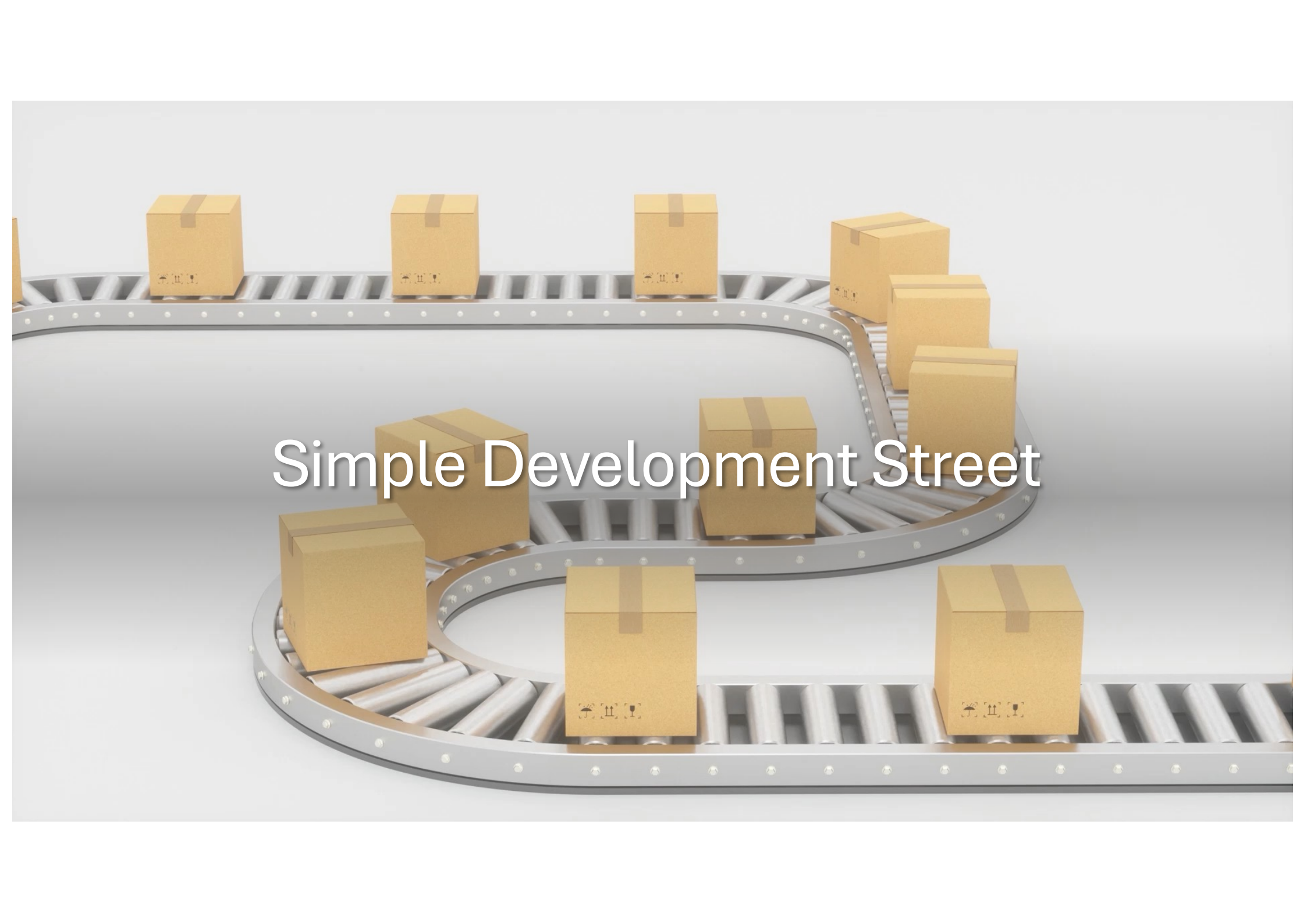




2	AUTHID CURRENT_USER		2	AUTHID CURRENT_USER	
3	as		3	as	
4			4		
5	- /* Copyright 2015-2023 - APEX RnD - United Codes		5	+ /* Copyright 2015-2024 - APEX RnD - United Codes	
6	*/		6	*/	
7			7		
8	--## CONSTANTS		8	--## CONSTANTS	
***** ↓ *****	@@ -156,6 +156,7 @@ c_special_ir_saved_report constant varchar2(19 char) := 'ALWAYS_REPORT_ALIAS';				
156	c_special_ir_repeat_header	constant varchar2(13 char) := 'REPEAT_HEADER';	156	c_special_ir_repeat_header	constant varchar2(13 char) := 'REPEAT_HEADER';
157	c_obfuscate_data	constant varchar2(14 char) := 'OBFUSCATE_DATA';	157	c_obfuscate_data	constant varchar2(14 char) := 'OBFUSCATE_DATA';
158	--### Debug		158	--### Debug	
159	c_debug_remote	constant varchar2(3 char) := 'Yes';	159	+ c_debug_no	constant varchar2(3 char) := 'No';
160	c_debug_local	constant varchar2(5 char) := 'Local';	160	c_debug_remote	constant varchar2(3 char) := 'Yes';
161	c_debug_application_item	constant varchar2(9 char) := 'APEX_ITEM';	161	c_debug_local	constant varchar2(5 char) := 'Local';
162			162	c_debug_application_item	constant varchar2(9 char) := 'APEX_ITEM';
***** ↓ *****	@@ -258,6 +259,7 @@ g_app_id number := null; -- APEX application id				
258	g_page_id	number := null; -- APEX page id	259	g_page_id	number := null; -- APEX page id
259	g_user_name	varchar2(200 char) := null; -- APEX user name (APP_USER)	260	g_user_name	varchar2(200 char) := null; -- APEX user name (APP_USER)
260	g_force_create_apex_session	boolean := false; -- Force creating a new APEX session	261	g_force_create_apex_session	boolean := false; -- Force creating a new APEX session
261			262	+ g_caller_program	varchar2(2) := null; -- Either null or DA (dynamic action)
262	--### APEX Page Items		263		
263	g_apex_items	varchar2(4000 char):= null; -- colon-separated list of APEX it ems e.g. P1_X:P1_Y, which can be referenced in a template using {Pxx_ITEM}	264	--### APEX Page Items	
			265	g_apex_items	varchar2(4000 char):= null; -- colon-separated list of APEX it ems e.g. P1_X:P1_Y, which can be referenced in a template using {Pxx_ITEM}
***** ↓ *****					



Demo



Simple Development Street



Dev

Prod

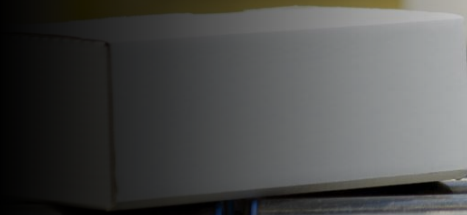


Dev/Prod



Rules of the Road

Main in your dev
environment must
be what is running
in production

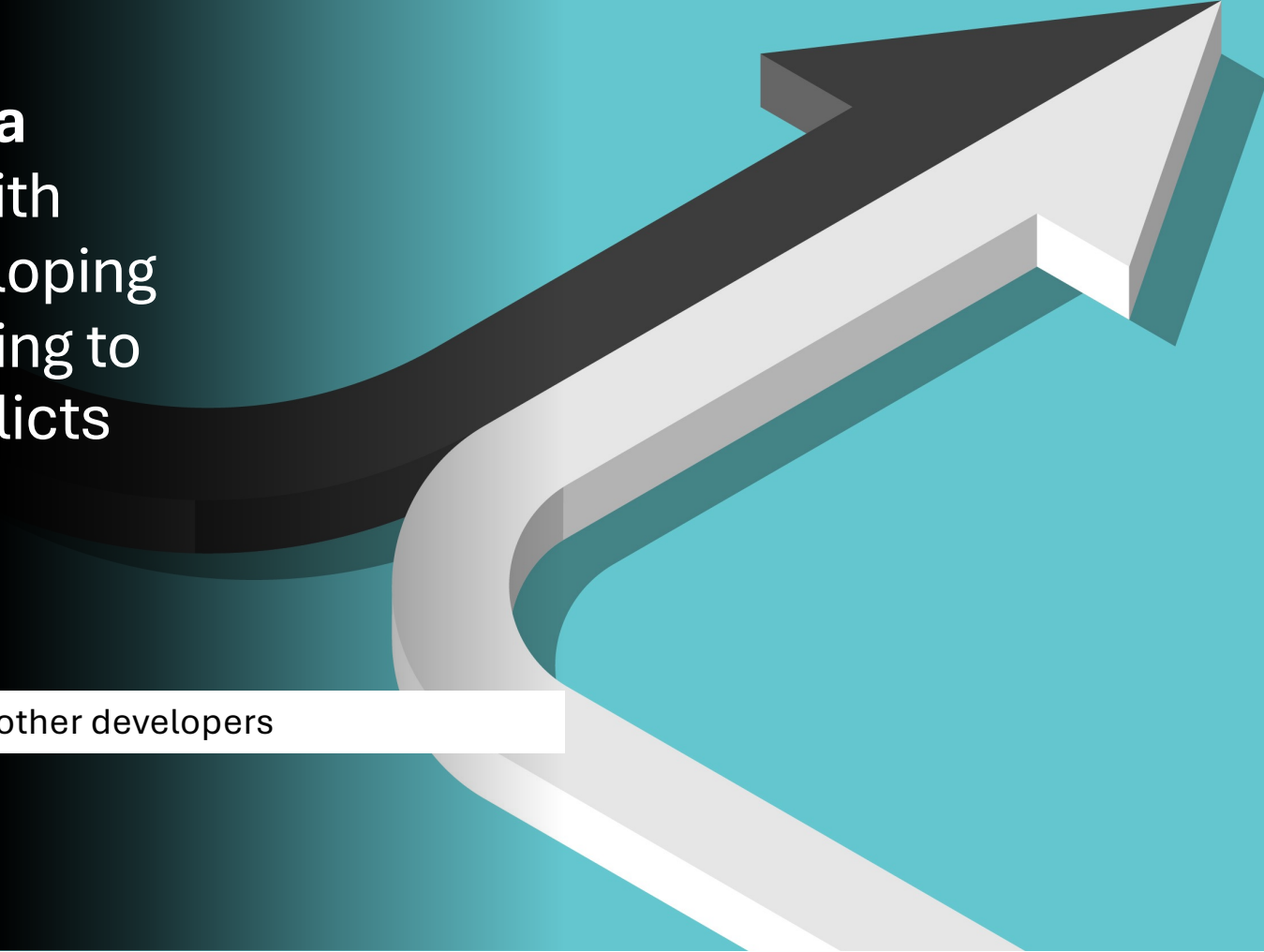


Working
copies should
be created for
each feature
or bug fix



**Always refresh a
working copy with
Main while developing
and before merging to
resolve any conflicts**

Select only changes from other developers



Working copies can't be moved (as working copies) from one environment to another, so merges must happen in dev

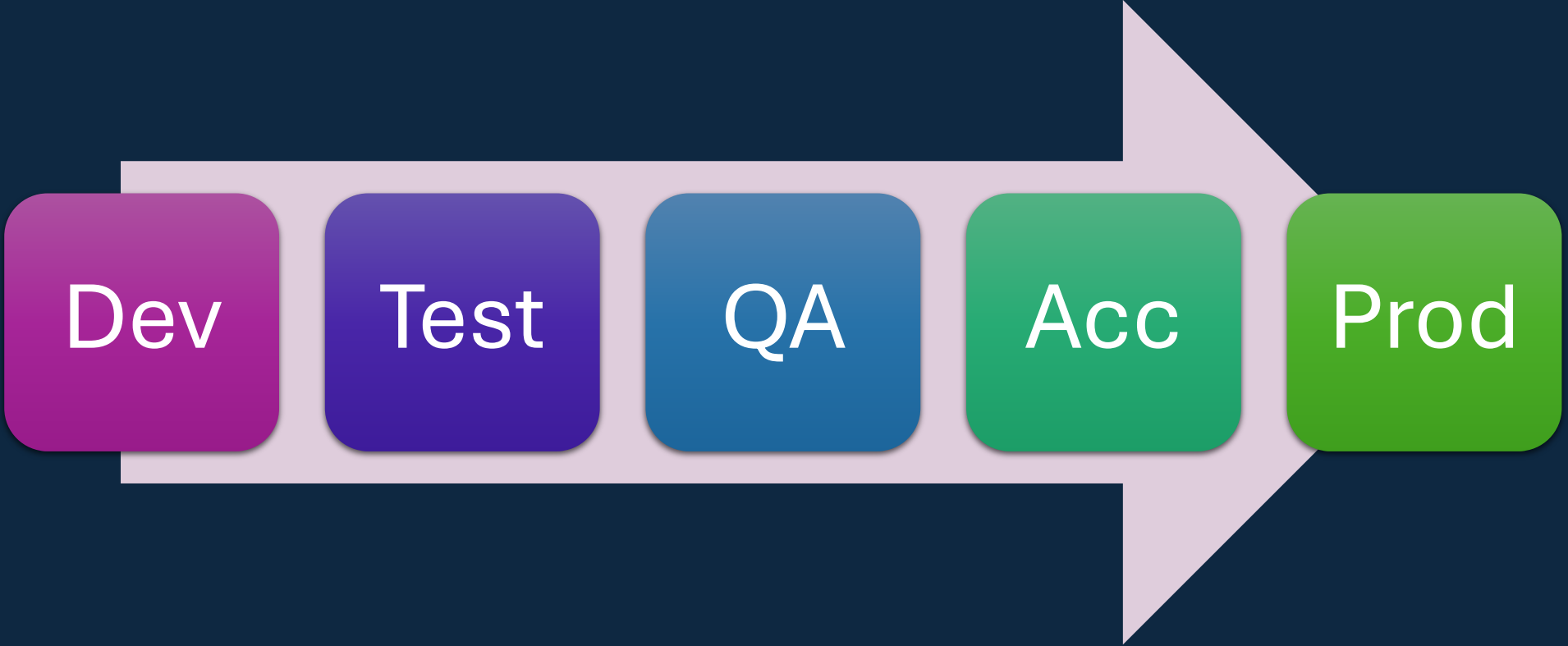


Once merged in dev
(and tested), the
patched main
application can be
moved to
production



Complex Development Street





Dev

Test

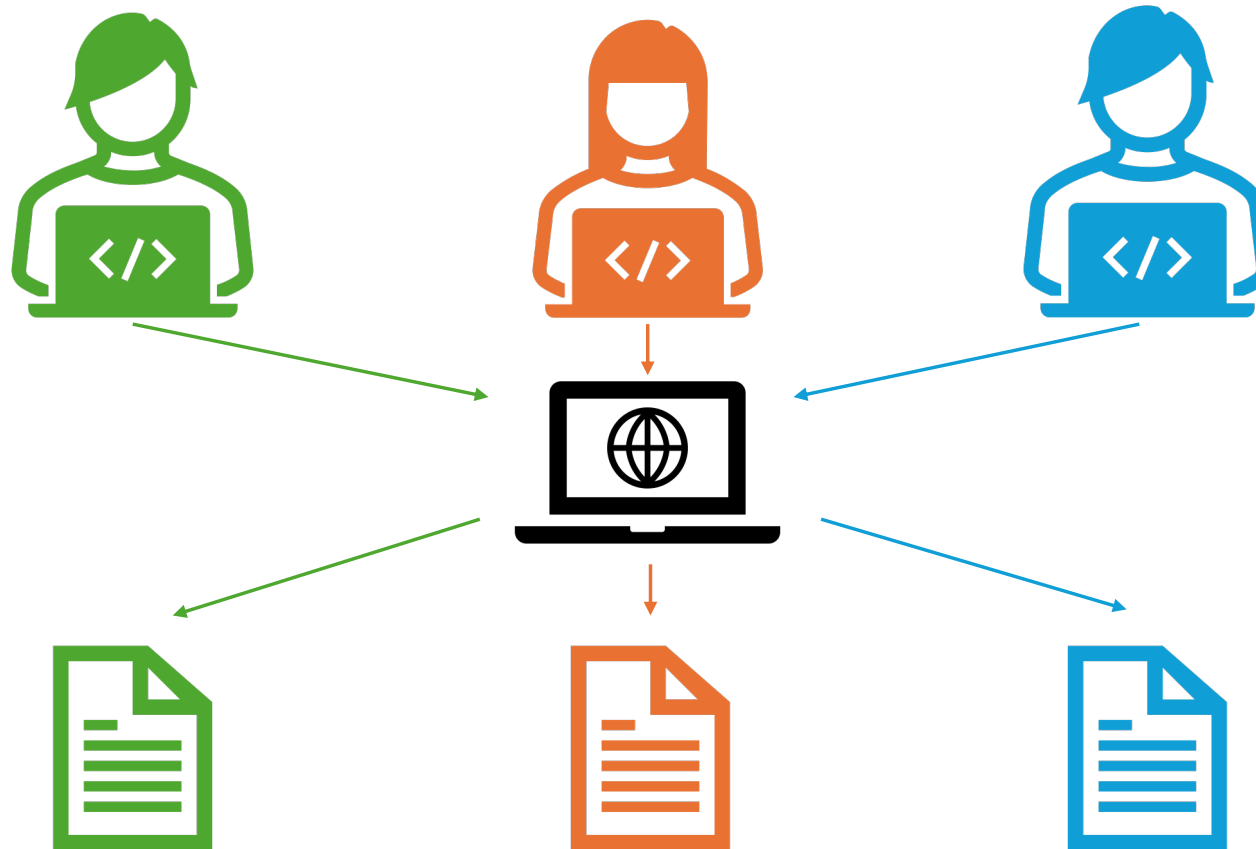
QA

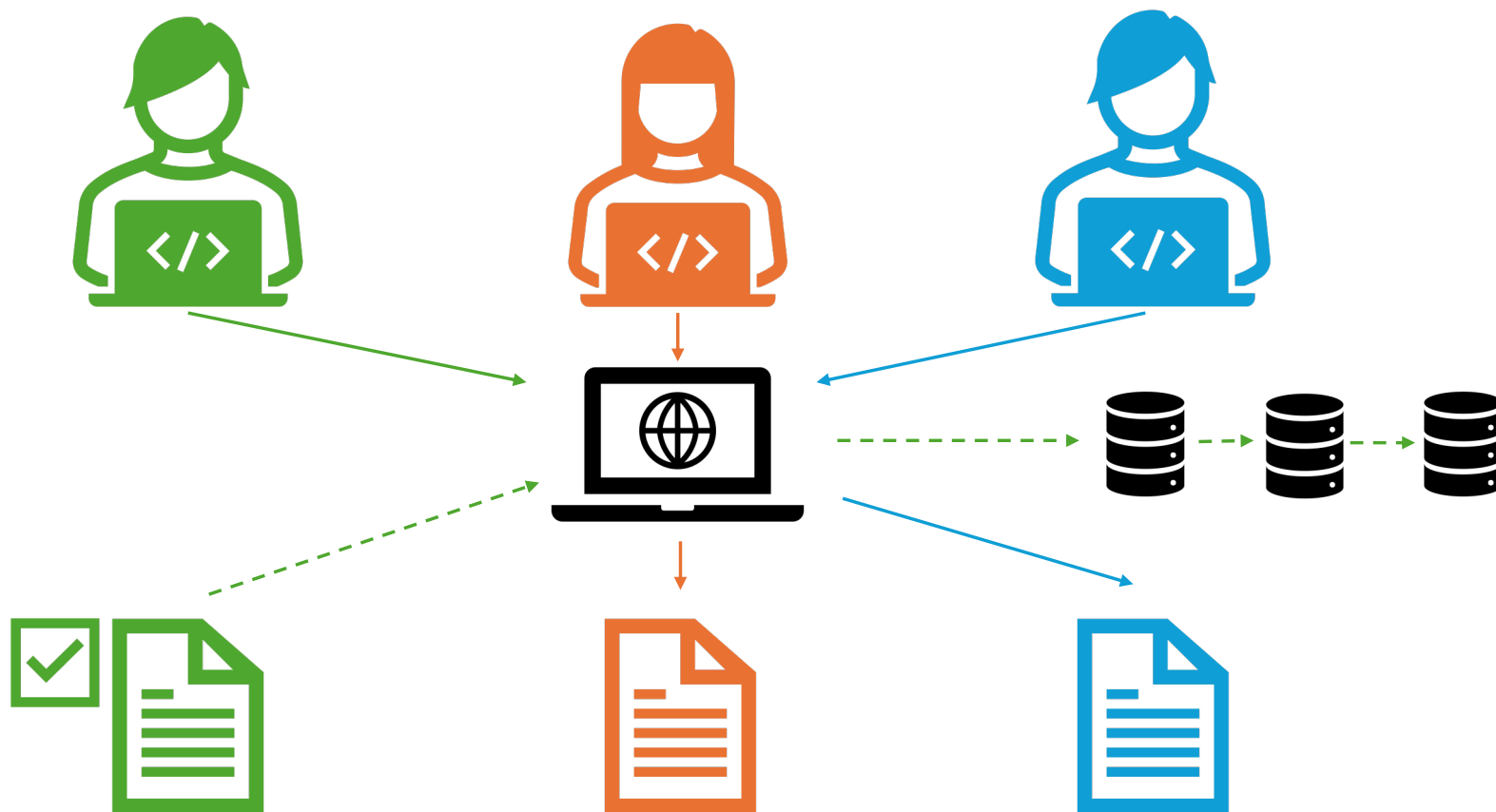
Acc

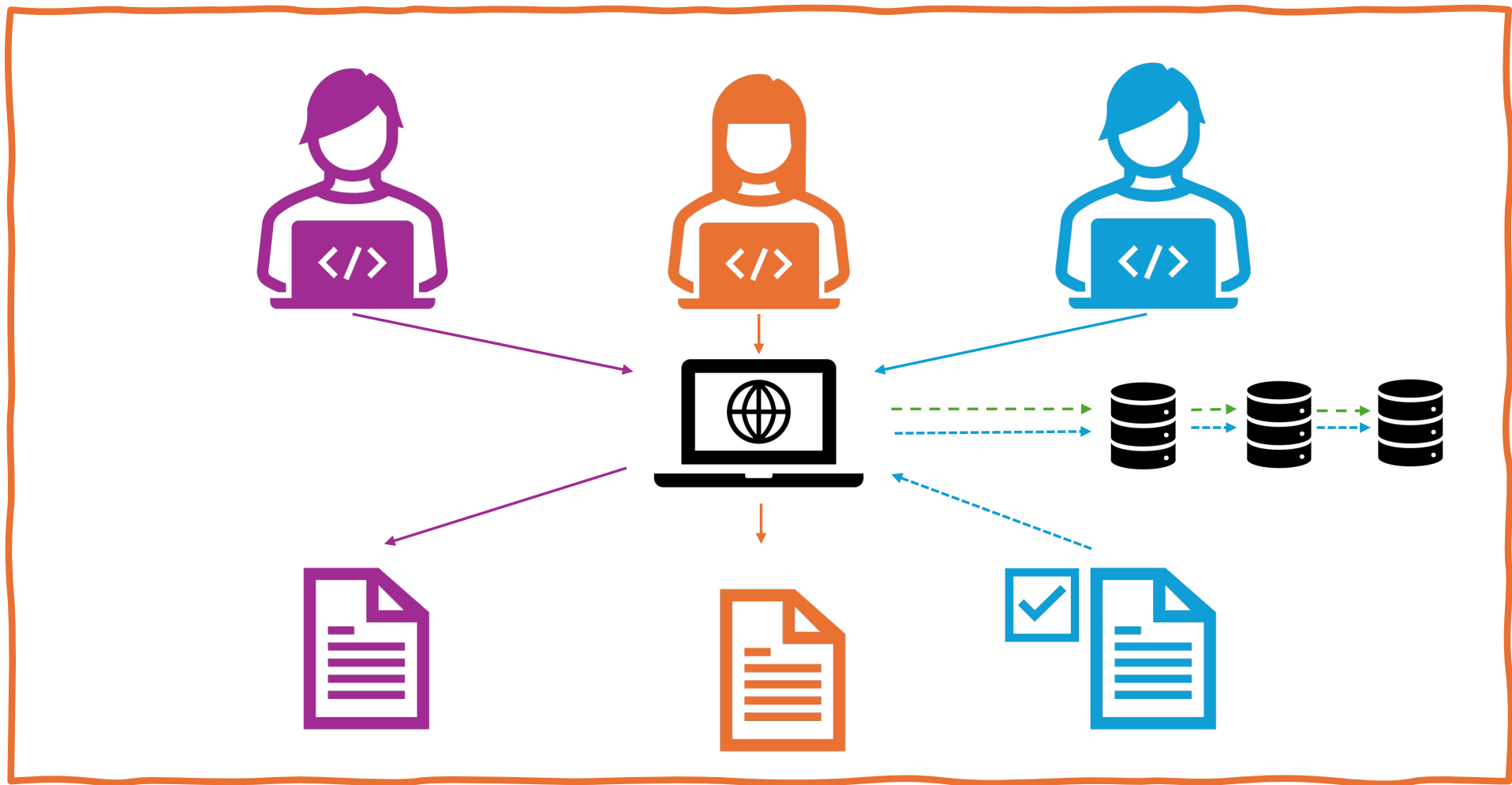
Prod

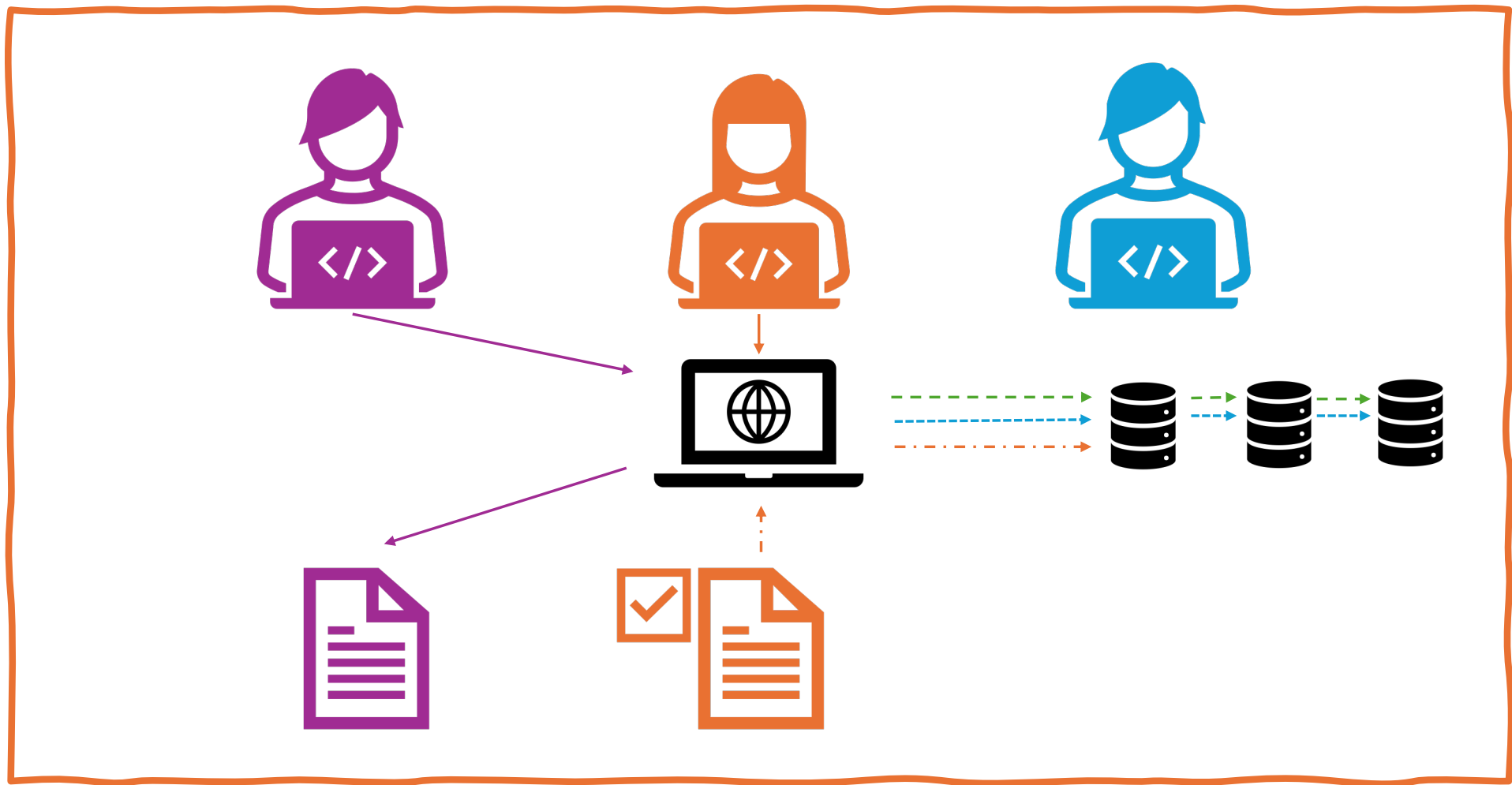
A “Simple” Scenario

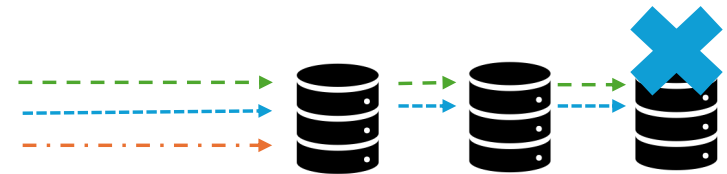
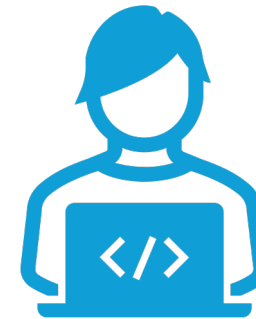
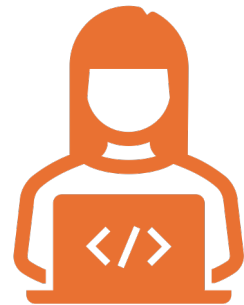
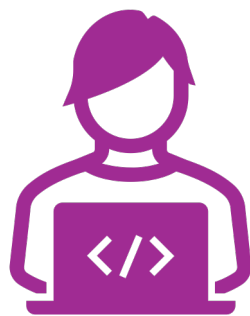




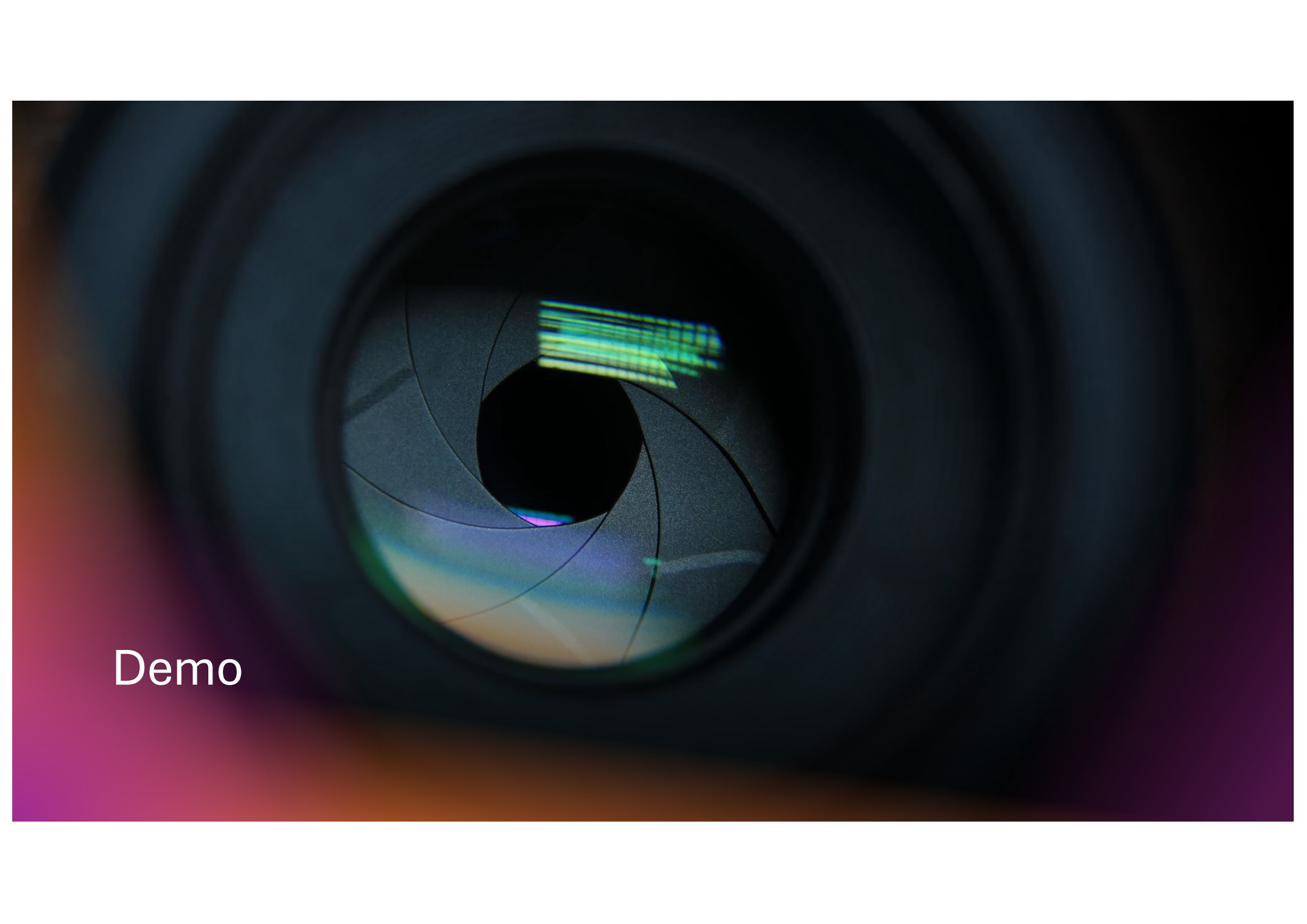








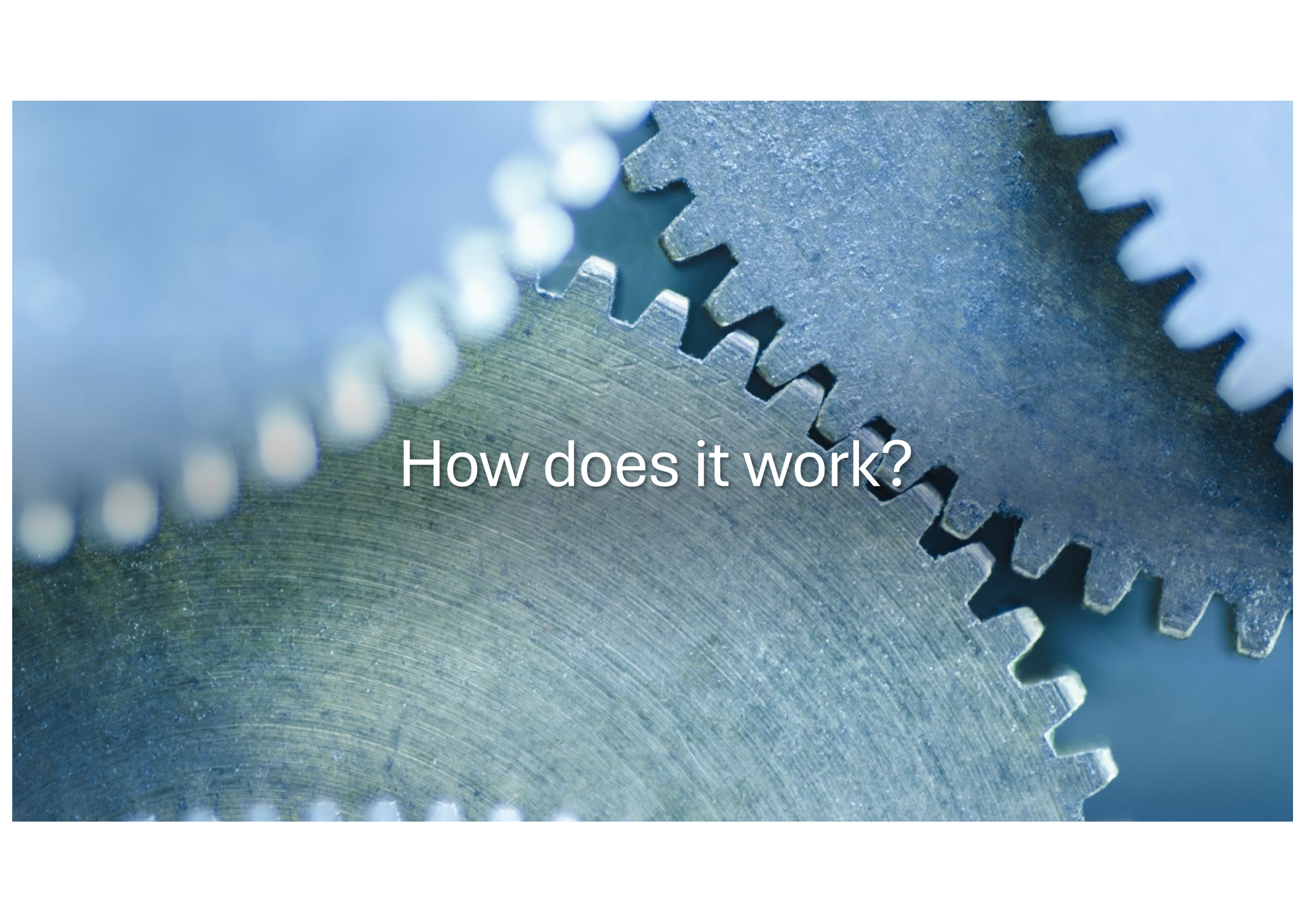
How do you fix **production** without introducing features in **test**?



Demo



Chaos ensues...

A close-up photograph of several interlocking metal gears. The gears are made of a dark, textured material, possibly steel, and are arranged in a way that shows their teeth meshing together. The lighting is dramatic, with strong highlights on the edges of the teeth and deep shadows in the gaps. The background is a solid, light blue color. The text "How does it work?" is overlaid in the center of the image in a white, sans-serif font.

How does it work?

What's missing?



Unsupported Component Types



Working Copies do not support the automatic merging of the following component types:

Translations

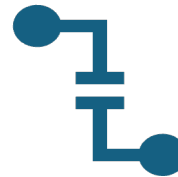
Themes

Theme Templates

Supporting Objects

Application Properties

Workspace-level components (such as App Groups, Credentials, and Remote Servers)



For unsupported component types, Oracle suggests developers merge the working Working Copy, delete it, and then manually apply the additions or changes for the unsupported component types.

In some cases, this might mean that you are making changes directly in production.

Limitations of Working Copies



You cannot create a Working Copy of a Working Copy.



Deleted working copy pages or Shared Components must be manually deleted in the Main application.



Components listed as Missing may have been removed in the Working Copy or may only exist in the Main application.

Merging is done at PAGE level

- Because the component export feature is used behind the scenes, we are stuck with page-level merging
- Changes by multiple developers to the same page will be overwritten when merging, causing manual intervention to resolve the conflicts.
- It would be nice to see when a region or item is changed



Merge Conflicts

If a component is changed in both main and working copy, there is no indication that a change was made on both sides; it appears like a normal change as if only one was updated.

With APEX 24.1, an indicator is present on a page if a change is being made in another working copy.

What do you want to see when a shared component is changed?

An LOV that is used on many pages

- Do you want to see changes on 10 pages or 1 change listed in shared components?

An authorization scheme could be on all pages

- Do you want to see changes on all pages or 1 change listed in shared components?

Not for large applications or Dev Streets... yet

- It could work for teams with very good communication, but rouge or uninformed developers can cause chaos.
- Developers need to be aware of their changes and the changes of others.
- Teams should have a strategy for when unsupported component types are changed.
- Teams need a strategy for patching production.
- **Working copies are great for small dev streets and teams with good communication**



Questions

