



SDL Tridion One-CMS vs DTAP

Tridion Expert Summit 2020

Lucas Liu

Senior Full Stack Developer in NetApp

BIO: 10+ years in SDL Tridion Cloud Implementation, DXA
Development, Data Integration and Migration



What is DTAP?

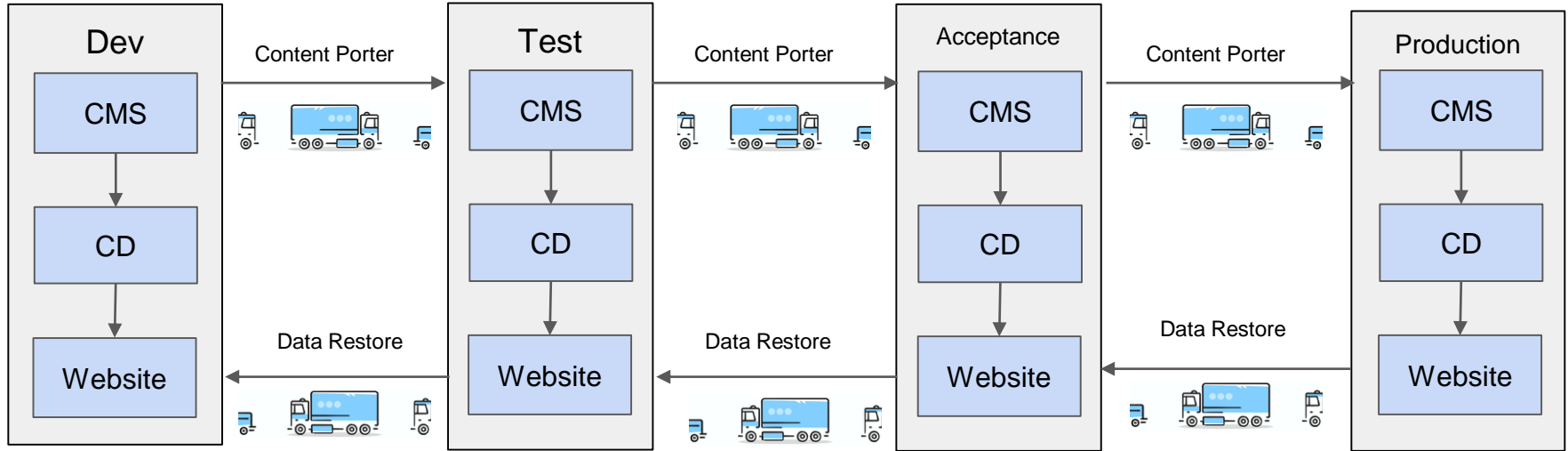


SDL Tridion DTAP traditionally including 4 separate environments (silos): Development Environment, Testing Environment, Acceptance Environment and Production Environment. each environment including a full set of CMS and Content Delivery.

For more information please follow the link for Flow in the DTAP setup
<https://docs.sdl.com/792152/536217/sdl-web-8-5/flow-in-the-dtap-setup>



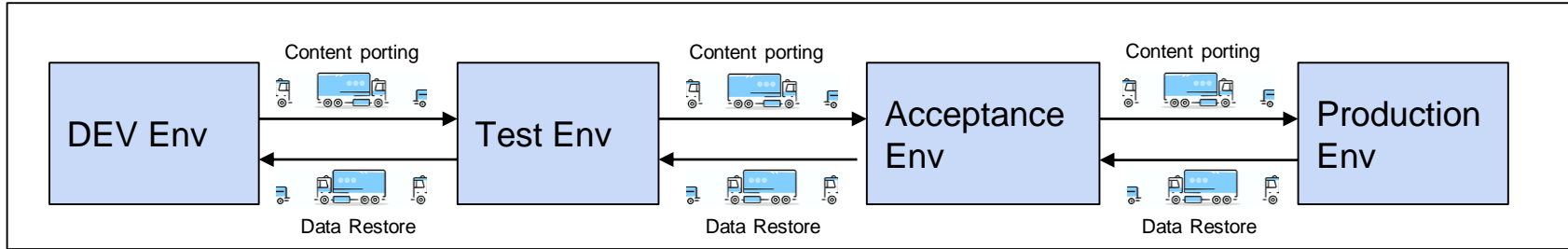
SDL Tridion traditional DTAP workflow looks like this:



A NEW Project generally starts from Dev environment, creating new schema and templates etc. When development is ready to test, using Content Porter migrates the items from Dev to Test environments for manual or automated testing. When Test is ready, Content port the items to Acceptance environment for final Review, After approval content port the items to Production to release to public. Data Restore back from Production to Acceptance, Test and Dev is also needed for verification and debugging with the same production data.



What are the DTAP Workflow Issues

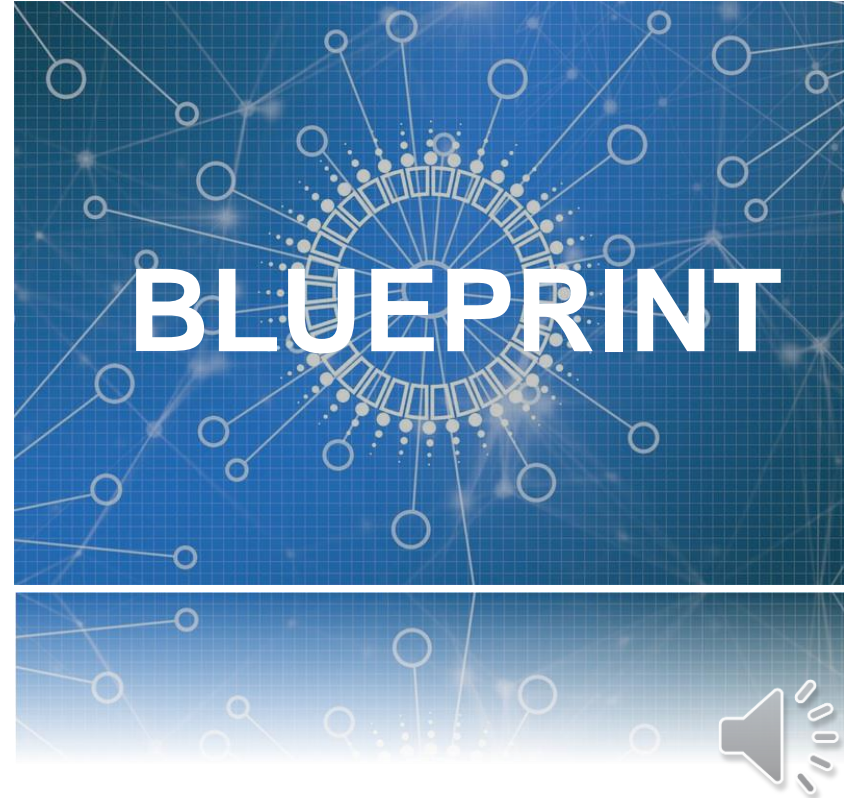


- Frequent content porting from Dev to Test to Acceptance to Production is required
- Frequent data restore from Production to Acceptance to Test to Dev is also needed for verification, debugging and troubleshooting
- There are Known issues in Content porting such as lost publishing target
- Also Known issues after Data Restore from other environments cause data inconsistency
- There always time cost in content porting and data restore
- Each DTAP environment require extra cloud resource, maintenance and troubleshooting

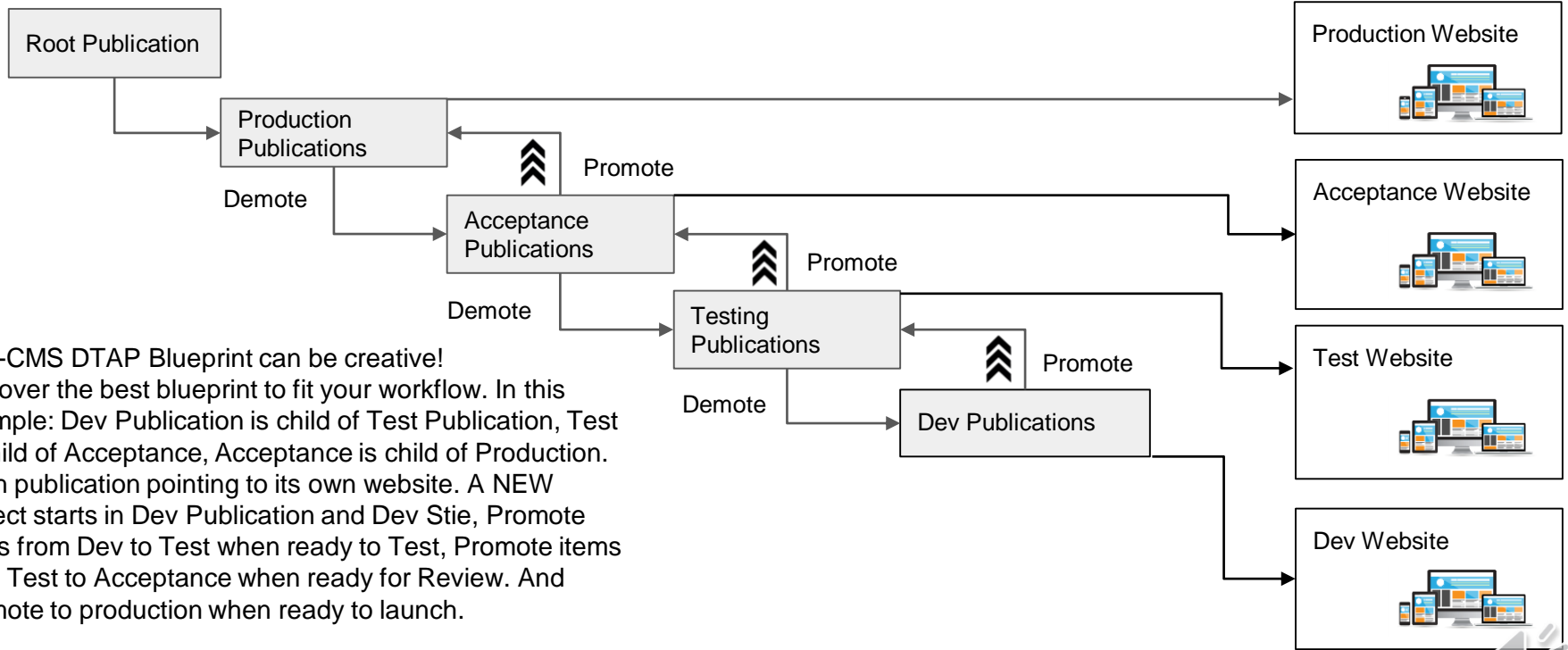


Traditional DTAP is dead, One-CMS DTAP Blueprint is emerging

One-CMS DTAP blueprint simulate DTAP workflow with **Blueprint** and **Promote/Demote** feature in SDL Tridion CMS. **Promoting** means that the content item is moved to a Publication higher in the Blueprint hierarchy; **Demoting** means the content item is moved to a lower Publication and removed from its current Publication. This is feature is ideal to simulate DTAP environments. **How to create DTAP environments in One-CMS can be creative!**



One-CMS DTAP Blueprint Example 1

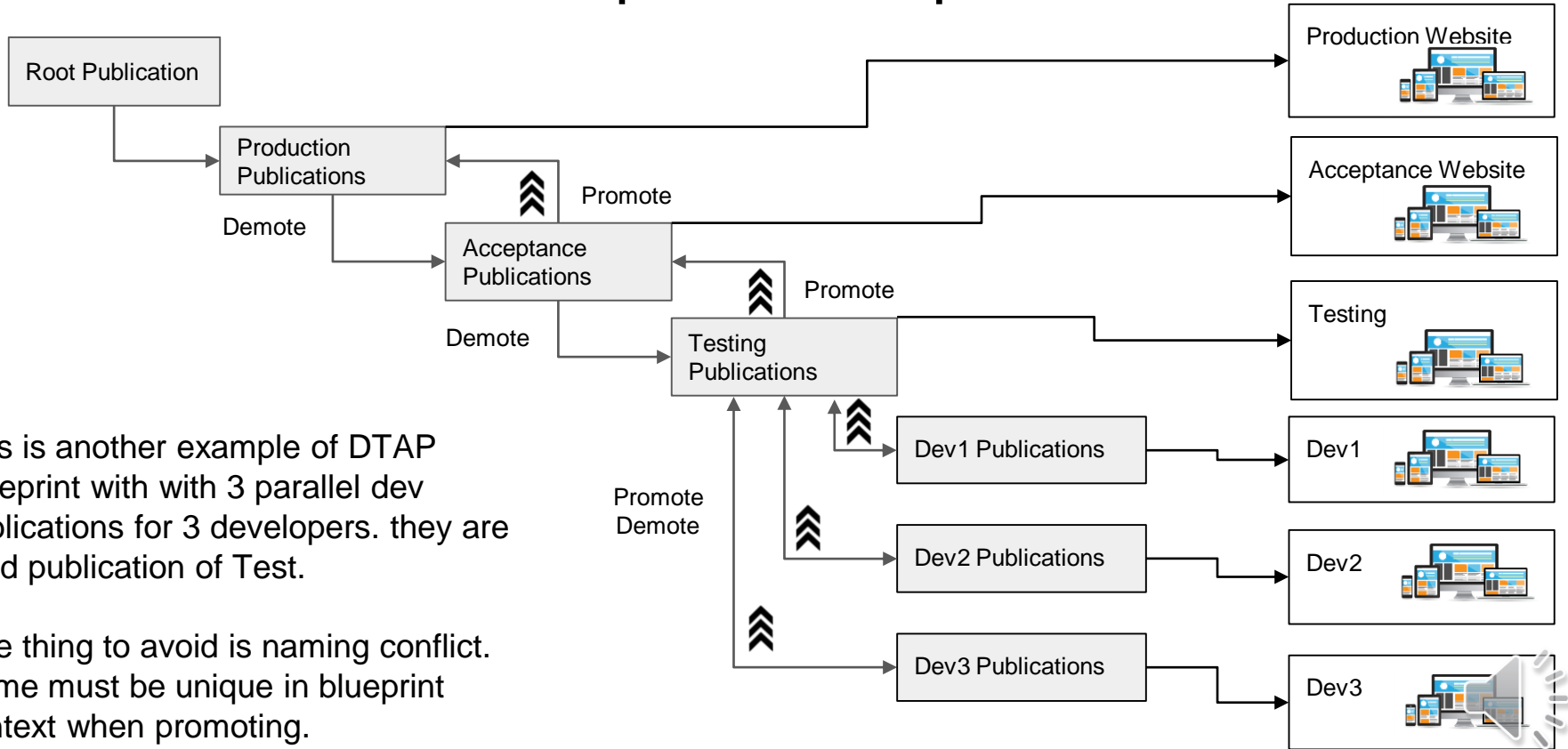


One-CMS DTAP Blueprint can be creative!
Discover the best blueprint to fit your workflow. In this example: Dev Publication is child of Test Publication, Test is child of Acceptance, Acceptance is child of Production. Each publication pointing to its own website. A NEW project starts in Dev Publication and Dev Stie, Promote items from Dev to Test when ready to Test, Promote items from Test to Acceptance when ready for Review. And promote to production when ready to launch.

The items promoting in blueprint is much easier than content porting and data restore.



One-CMS DTAP Blueprint Example 2



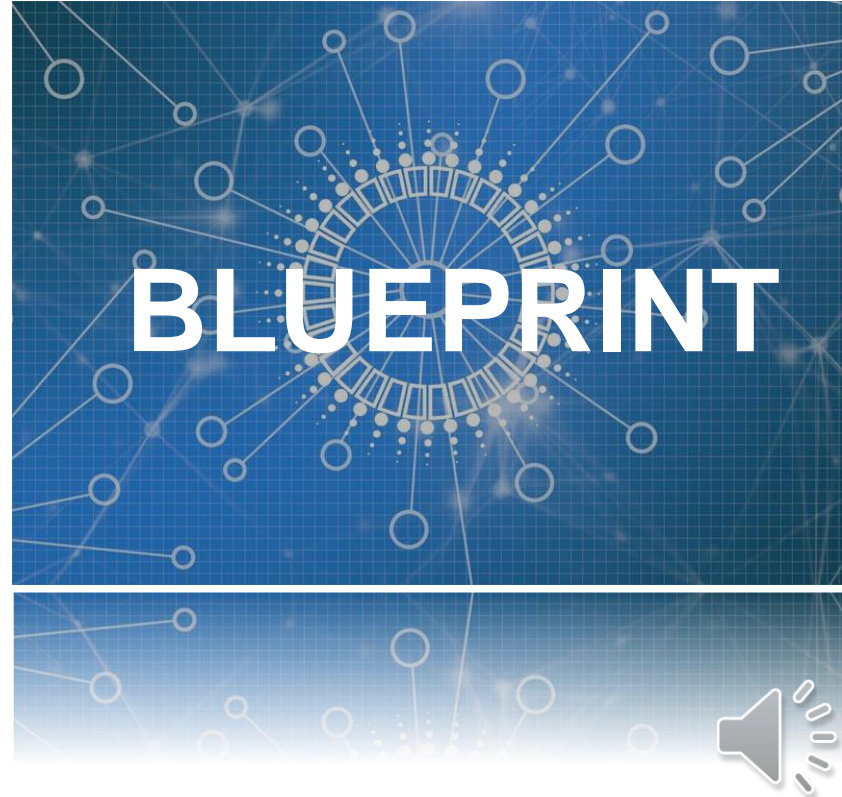
This is another example of DTAP Blueprint with 3 parallel dev publications for 3 developers. they are child publication of Test.

One thing to avoid is naming conflict. Name must be unique in blueprint context when promoting.

At the end: One-CMS DTAP Blueprint benefits are

- No content porting needed
- No data restore needed
- Data always in sync in the same blueprint context
- Promote/Demote process is fully manageable for any conflict
- Flaxible DTAP environment to fit your workflow
- Cost saving in cloud resources for Acceptance, Test and Dev CMS

As result, One-CMS DTAP Blueprint merged SILOs, simplified workflow, shorter the time to market, and lower the Total Cost of Ownership.



QA

thank
you!

