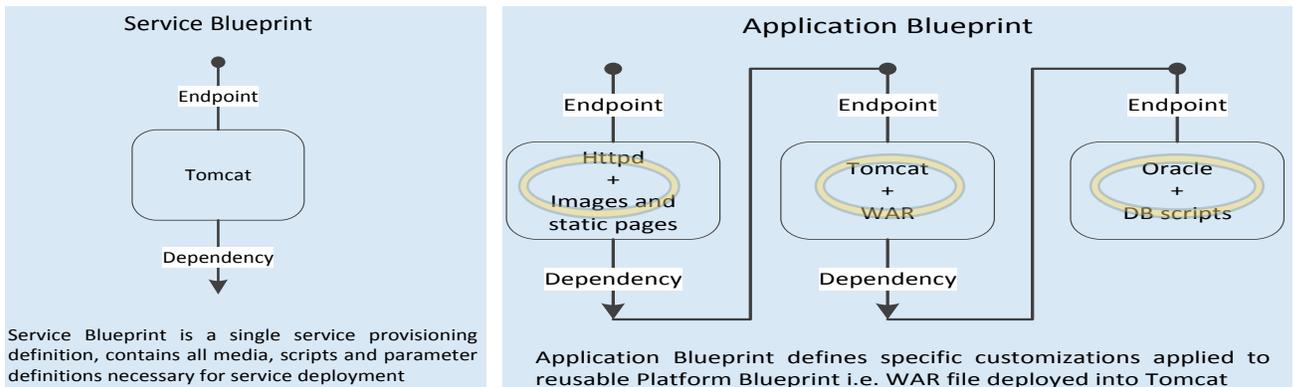


UNITY APPLICATION MANAGEMENT

Unity Application Management has been designed to hide complexity of the infrastructure and to provide Application Blueprints that work in similar fashion on any platform or cloud. In this case, the application Blueprint composition becomes a simple mechanism that bridges the differences between clouds and defining the know-how of multiple clouds operation. Hence, the Applications that have been deployed on VMware and Bare-metal, could be easily migrated to AWS or Azure. The ITs do not have to re-build the Applications, while making cloud adaption an easy process and turning Application Management into a great productivity and value-add tool.



Besides automated application management, Unity allows on-demand generation of images. In this case, Unity automatically generates and “caches” any image while “sees” it first time in any Datacenter, Region or a Cloud. Moreover, the Unity orchestration model is adapted to using automatically these “optional” images. For example, instead of creating 100 installations, it would clone the images 100 times to replicate anywhere across given environment. But this is not all. Unity can perform this orchestration for a mixed environment, e.g., VMs and Containers. It can automatically build container “gold image” behind the scene, while requiring no manual management. Same Blueprint can be used to build either Container or VM image. As a result, when it is ready, the IT organization can seamlessly add Container deployment to production without changing environment and extra testing.

Key Principals

From	To
Diagrams and specs	Visual Composition
Snowflakes with exponential complexity	Heavy reuse, pattern based
Explicit Infrastructure Management	Invisible Infrastructure
DevOps coding for every app	Very little to none application specific DevOps code
Choose between image cloning systems or dynamic node configuration systems (Chef)	Images are automatically generated as a cache but anything can be built from scratch. Docker included.
Complexity of change, upgrade and patch	Built-in upgrade and patch model with testing
Disconnected tooling	Integrated with CI/CD, dev., QE and AppOps