TECHNICAL WHITE PAPER

Pure Fusion

Self-service Storage-as-Code™

Contents

Introduction

This brief introduces Pure Fusion, a new offering from Pure Storage® that unifies FlashArray™ systems into a policy-driven service with a cloud-like Storage-as-Code™ management and consumption model. Pure Fusion simplifies storage management for:

Providers1 (IT Departments): By consolidating FlashArrays into cloud-like availability zones (AZs) that scale up or out, standardizing configurations, enabling self-service provisioning while controlling users' access to capacity, and rebalancing I/O loads as needed.

Consumers (applications): By presenting classes of storage from which virtual storage resources[2] can be allocated and self-managed using data center automation tools or user-developed software.

What's the Problem?

In modern data centers, managing processing, network, and storage resources and responding promptly to the needs of multiple consumer constituencies is the challenge for IT teams. Beyond capital equipment and software costs, managing a constantly changing IT landscape requires substantial human skill and energy. This is especially true in the case of storage, where it's not only about hardware, but also about valuable data assets that must be kept accessible to consumers and protected commensurately with their value.

The Cloud to the Rescue?

The challenges of managing on-premises storage have undoubtedly been a factor in the widespread adoption of public cloud storage. In addition to replacing capital expenditures with more flexible operating models, public clouds shift the burdens of managing equipment and configuring storage resources from the consuming organizations to cloud providers.

Public clouds simplify storage management for IT departments, but more importantly, they fundamentally change how users perceive and consume storage:

- They expect to acquire and relinquish capacity with specified performance and protection properties at a moment's notice, often using data center automation software.
- They are comfortable with virtualization, requiring only that cloud providers meet service level agreements (SLAs).
- They realize that their resources share hardware with other consumers, but they expect them to be logically isolated from those of other consumers.

[1] This brief refers to IT operations departments as providers and parts of an organization (e.g., divisions, departments, functional teams, etc.) that use digital data storage as consumers.

[2] This brief refers to virtualized storage capacity structured as volumes (and eventually file systems, and object stores) as storage resources. In the FlashArray context, volumes are storage resources.

The Cloud is Not for Every Data Set

While public cloud storage has advantages, there are downsides, as well:

- The low cost of storing "busy" data is quickly overwhelmed by network and access charges.
- Cloud I/O performance is inadequate for the most I/O intensive applications.
- In organizations with many consumer departments, costs can be difficult to control.
- Organizations may be reluctant to entrust sensitive data to public clouds, or regulatory constraints may prohibit them from doing so.

For these reasons, many public cloud users also maintain significant amounts of on-premises storage, often in multiple data centers. On-premises storage eliminates access charges, delivers high I/O performance, and provides the sense of security that comes from knowing where key data sets are stored and how they are protected.

But on-premises storage, the data it contains, and constantly changing consumer requirements can be difficult for IT teams to manage effectively, especially in organizations that serve many consumers across multiple data centers. These organizations would be better served with a cloud-like storage paradigm, both in public clouds and on-premises, to enable their consumers to self-manage resources while adhering to organization policies.

What's Wrong with Today's Way of Doing Things?

Historically, storage systems and data have usually been managed by the same team. With a single data center and a handful of consumers, this makes economic and technical sense. But for organizations with hundreds of applications and petabytes of data, there are too many data centers, systems, hosts, applications, and data sets, all changing too rapidly, for "storage management by spreadsheet" to be effective. Consolidated storage management and consumer self-service are practical necessities and provide consistent access, adequate I/O performance, robust security, and protection for data assets stored on-premises.

Today, a typical IT team must be fully aware of *what* data is stored, *which* storage systems contain it, *how* it is protected, and *who* has access to it at any point in time. Likewise, consumers must be aware of the physical locations and properties of the systems that contain their data.

At scale, this model requires a full range of knowledge, skills, and situational awareness from both IT teams and consumers. Consumers submit requests for storage resources and wait for IT teams to fulfill them. IT teams attempt to satisfy whatever requirements consumers specify, often resulting in inconsistent treatment of data assets. Managing storage in this way means slow response to requests for new resources and changes to existing ones, inconsistent performance, inadequate protection, and inefficient utilization of storage capacity.

The Pure Fusion Difference

Pure Fusion streamlines the management of on-premises storage with a consumer self-service model that isolates consumers from storage system, connection, and data protection details. It provides organization-wide availability, performance, and connectivity standards for storage resources and enables consistent protection of the data they contain.

Managing on-premises storage involves both providers and consumers, each with its own responsibilities. With public cloud storage, the cloud vendor is the provider. Consumers allocate and manage virtual storage resources using vendor-provided menus. On-premises, however, the organization is both provider and consumer—IT teams provide storage capacity and consumers negotiate for storage resources whose properties meet their applications' needs.

The first key Pure Fusion innovation is separation of these roles. Providers are responsible for:

- Installing, configuring, and managing arrays
- Defining the storage and data protection options available to consumers
- Managing the organization's storage consumers

Consumers acquire and relinquish their own storage resources using a cloud-like self-service model. They use storage class templates defined by providers to specify resource properties. Each consumer's resources are logically distinct from those of other consumers.

The second key Pure Fusion innovation is a "storage as code" design center. Each of its provider and consumer facilities, corresponds to a REST API. The initial release includes a provider GUI for infrequently performed tasks, consumers create and manage their resources by embedding API calls in data center automation suites (e.g., Ansible) or in user-developed software tools.

When enabled by an array administrator, Pure Fusion replaces local control with consolidated management invoked via the Pure1® cloud. As a cloud service, Pure Fusion is highly available, always up-to-date, and requires no user maintenance. It is a management facility ("control plane") that uses the Pure Edge Service (PES)[3] for communication between managed arrays and the Pure1 Cloud. PES is not in the host-array data path, however. Temporary loss of access to the Pure1 Cloud does not affect an array's ability to execute hosts' I/O commands.

Pure Fusion and The Storage Management Experience

Pure Fusion improves on-premises storage management in four important ways:

- **Self-service on-premises storage**: Storage consumers create and manage their own storage resources with no awareness of their location and with no provider involvement.
- **Organization-wide standardization**: With storage classes and data protection policies, consumers' storage resources are both consistent with organizational policies and commensurate with data set values.
- **Non-disruptive infrastructure management**: Providers can add to or replace the arrays managed by Pure Fusion, relocate storage resources, and adjust template and policy properties without disrupting applications.
- **Storage-as-Code**: Pure Fusion facilities are invoked via REST APIs, so in addition to Pure's own provider GUI, they can be embedded in user-developed software tools, data center automation packages, and container orchestrators.

[3] PES is described in TB-211001, available from Pure Storage representatives.

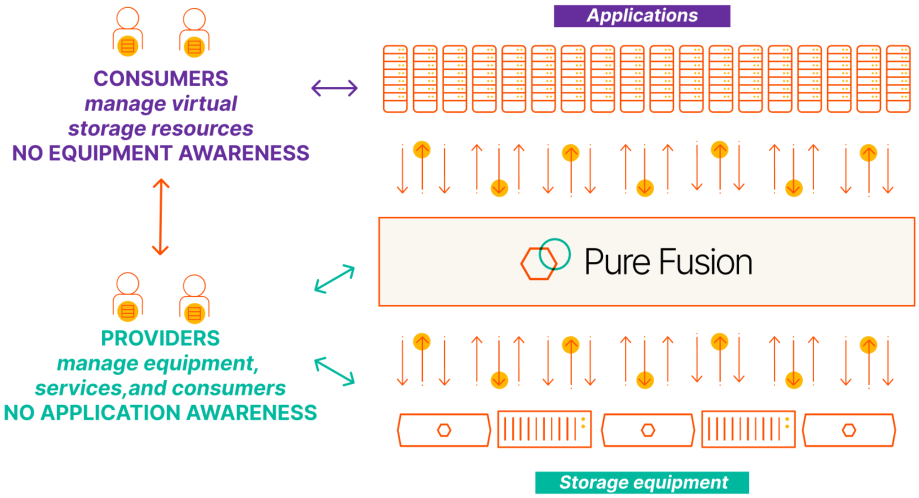

Figure 1: Storage management with Pure Fusion

Together, these innovations sever the direct connection between storage consumers and the physical infrastructure and make self-service on-premises storage possible. At the same time, they improve storage quality by with consistent resource properties and transparent rebalancing. In addition, they lay the groundwork for integrating storage management with other automation capabilities that may be offered by Pure or by other vendors in the future.

Pure Fusion Provider Responsibilities

Separating storage consumer and provider roles simplifies key provider tasks by making them transparent to consumers and non-disruptive to overall IT operations. Providers (typically IT operations teams) manage the infrastructure, the services provided to consumers, and the consumers themselves, but not consumers' virtual storage resources.

Infrastructure Management

In addition to acquisition, maintenance, and lifecycle management, all largely outside the scope of Pure Fusion, providers' responsibilities include:

- **Organizing arrays**: Assigning arrays to and removing them from the managed infrastructure.
- **Configuring Networking**: Assigning blocks of IP addresses and configuring network parameters to ensure that arrays can communicate with hosts that require access to the storage resources they contain.
- **Fault handling and load balancing**: Remediating hardware and software faults and relocating storage resources to rebalance load. Both functions are transparent to consumers.

Organizing Arrays

Pure Fusion organizes managed arrays in a two-tier hierarchy with broad user acceptance throughout the industry. Providers assign each managed array to:

- **A region**: Conceptually, a grouping of geographically proximate arrays. For organizations with data centers in multiple locations, regions enable consumers to place resources close to the hosts that use them for optimal performance and availability and low communication cost.

- **An availability zone (AZ)**: Conceptually, a set of arrays comprising a storage cluster that can be scaled out as needed by adding arrays. The AZ is the unit within which rebalancing occurs. Pure Fusion places each new or relocated storage resource on the “best fit” array in the AZ specified by the consumer.

Figure 2 illustrates the region and AZ concepts. Provider-defined regions would typically include geographically proximate arrays; AZs might include the arrays in a data center row or rack.

In a world of fast, pervasive networking, ubiquitous flash memory, and an evolving scale-out application architecture, next-gen shared accelerated storage has the power to unite both networked and direct-attached storage in a single, shared architecture. A shared design consolidates data silos; accelerates production, DevOps, and data analytics; and helps enterprises pivot to a data-centric architecture.

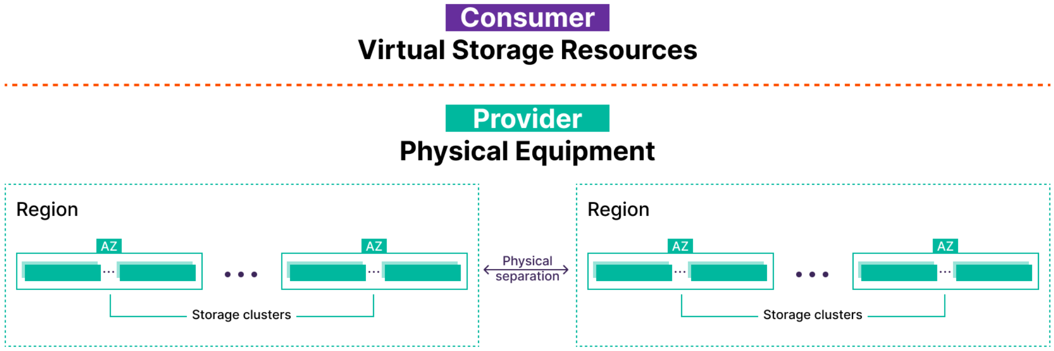

Figure 2: Regions and availability zones

Service Management

Providers specify their organization’s storage policies with storage classes and protection policies—sets of properties that consumers specify by name when creating storage resources and arranging for data protection:

- **Storage classes**: These set resource capacity and performance limits, and the array types (FlashArray models) on which resources may reside.
- **Protection policies**: In the initial release, these are snapshot schedules for protecting resource contents against loss and corruption. Other capabilities (e.g., replication) may be added in the future.

To create a storage resource, a consumer specifies a storage class, a capacity, and the AZ in which to locate it. Observing storage class constraints, Pure Fusion creates the resource on the array best suited to satisfy the consumer’s request. Consumers can apply protection policies to resources when they are created or at any time thereafter.

Consumer Management

Providers use a two-tier structure of tenants and tenant space objects to control consumer access to storage. Conceptually, tenants are an organization’s users of IT resources, such as divisions or departments. A Pure Fusion tenant object consists of:

- One or more tenant administrator accounts
- One or more tenant spaces, each with its own tenant space administrator(s).

The tenant concept provides coarse control over managed storage. Tenant administrators exercise finer control with self-administering tenant spaces, typically corresponding to applications. Tenant space administrators are the consumers of managed storage; each one creates and manages resources for its own application(s) in a management multi-tenancy model. Tenant space administrators cannot manage other tenant spaces' resources.

As an example, Figure 3 illustrates an organization with tenants representing product development and business management functions.

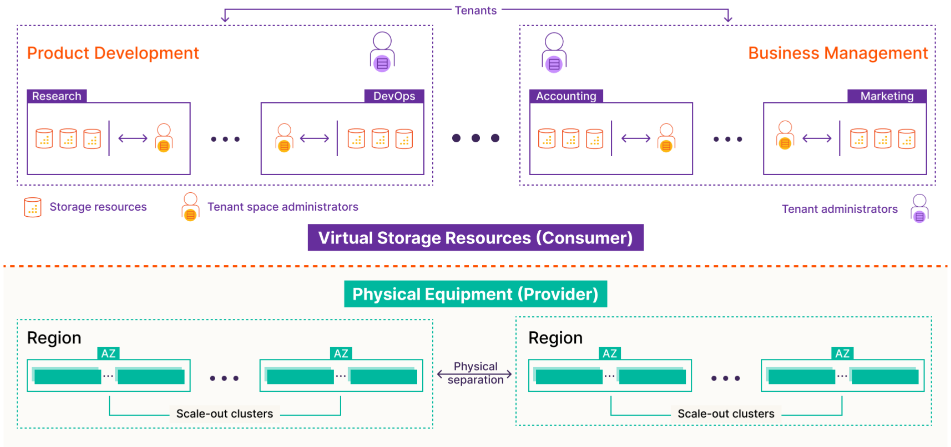

Figure 3: Tenants and Tenant Spaces: an example

In the figure, the product development tenant has tenant spaces for research, DevOps, and other functions. The business management tenant has tenant spaces for accounting, marketing, and more.

Each tenant space administrator manages storage resources for its own application(s). Tenants take a broader view in that they create separate tenant spaces as needed by applications within their areas of responsibility. A key advantage of Pure Fusion, however, is that neither tenants nor providers micromanage storage resources for individual applications.

Pure Fusion Structure

Pure Fusion is a distributed application whose overall structure is illustrated in Figure 4. The main component runs in the Pure1 Cloud, where it executes Pure1 user commands. Both providers (AZ administrators) and consumers (tenant and tenant space administrators) issue commands to the cloud component via the Pure1 GUI or via software containing embedded Pure Fusion APIs.

The cloud component executes user commands by converting the actions they specify (e.g., add array to AZ, create storage resource, etc.) into sequences of local APIs for execution by managed arrays.

Each managed array runs a Pure Fusion agent, which receives messages from the cloud component via the Pure Edge Service (PES) and issues the APIs they contain to the array.

Pure Fusion is an opt-in facility. Agents are part of the Purity operating environments shipped to customers. To opt in, a local array administrator must enable PES messaging and agent installation and updating. A Pure1 user with the Pure1admin role then downloads and installs the agent.

An array may run multiple PES agents, each installed on both controllers. Some, like the Self-Service Upgrade agent, are active on both controllers simultaneously. Others, like the Pure Fusion agent, run in active-standby mode. A FlashArray primary controller's Pure Fusion agent receives and executes commands; the secondary controller's agent becomes active only when controller failover occurs.

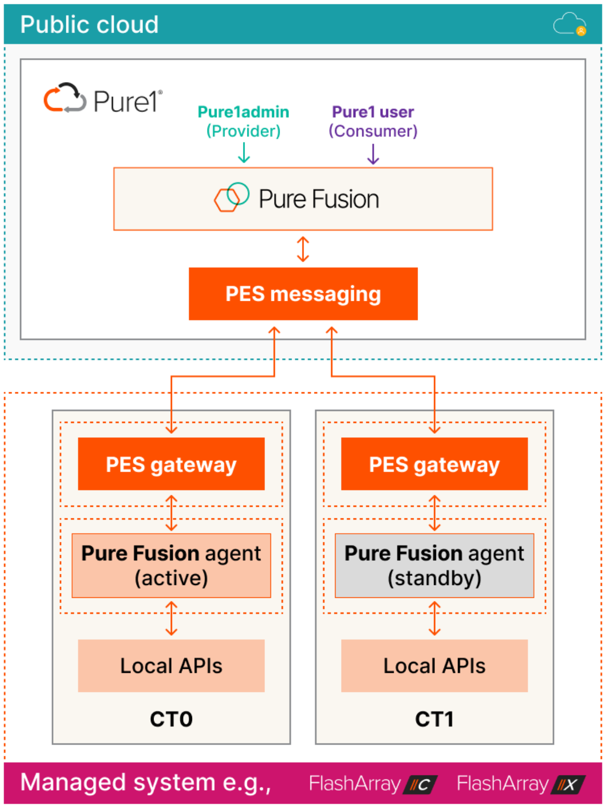

Figure 4: Pure Fusion Structure

Pure Fusion Objects

Pure Fusion is comprised of two sets of objects with one managed by providers and the other by consumers. The two are connected by provider-defined policies.

Provider Objects

Providers specify their organization's storage policies with storage classes and protection policies—sets of properties that consumers specify by name when creating storage resources and arranging for data protection.

Availability Zones

The most visible provider-managed object is the availability zone. An AZ is an extensible collection of arrays within which storage resource placement and rebalancing occur. The arrays that comprise an AZ may be of different types. For example, an AZ may contain FlashArray//XL, FlashArray//X, and FlashArray//C models.

An AZ is effectively a storage cluster that consumers perceive as a single system roughly analogous to a locally managed array. It is the universe within which storage resources are allocated and within which providers rebalance load by relocating workloads (sets of related resources) from one array to another.

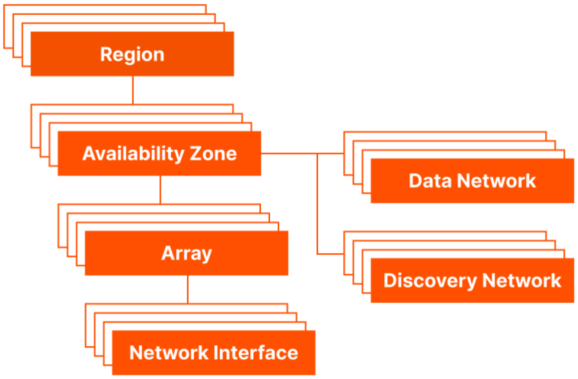

Figure 5: Provider Objects

Networks and Interfaces

For an AZ to appear as a single system to a consumer, all hosts must be able to communicate with all its arrays. But for an AZ administrator to rebalance load by relocating resources transparently, hosts must communicate with storage resources at the same addresses regardless of where the resources are located. Pure Fusion achieves this with a dual networking structure consisting of:

- **Discovery network**: A pool of virtual network addresses (e.g., VIPs in Ethernet-connected arrays) used by hosts to discover and address resources.
- **Data network**: A pool of storage network addresses dedicates to an AZ. When an array is added to an AZ, Pure Fusion assigns addresses from the pool to its network ports. It maps discovery network addresses to data network addresses dynamically as storage resources are created and relocated. With Ethernet networks, for example, iSCSI redirection maps an AZ's discovery network VIPs to IP addresses in its data network dynamically.

Figure 6 uses iSCSI to illustrate the AZ dual network concept. Pure Fusion assigns IP1…IP4 from its data network pool to the FlashArray network interface ports when the array is added to the AZ. When a storage resource created by a consumer lands on FlashArray, the array's iSCSI mapping tables are updated to redirect discovery network addresses VIP1…VIP4 to IP1…IP4.

When a Pure1 user rebalances AZ load by moving resources to different arrays, the relocated resources' IQNs are associated with the new array's ports, and the arrays' iSCSI mapping tables are updated to redirect the VIPs to the resources' new location. The iSCSI session connection to the relocated resources is then replaced by a new session between the host and the new location. None of this disrupts host access or I/O.

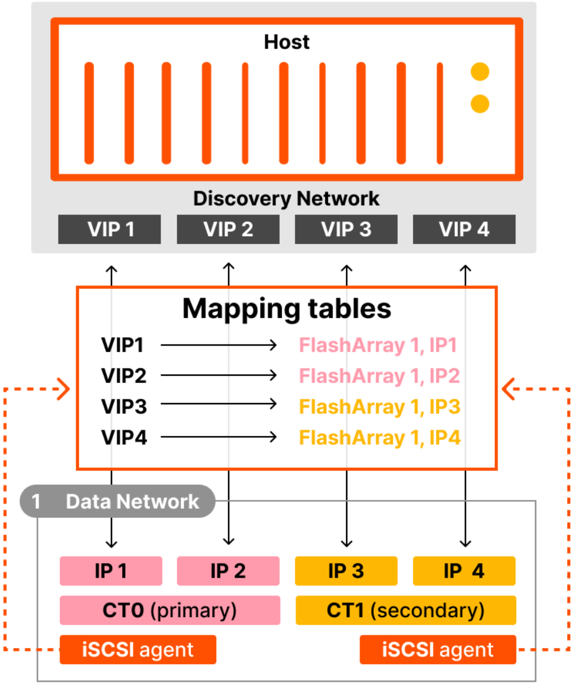

Figure 6: Discovery and data networks

Policies

With Pure Fusion, storage providers create and manage policies that consumers specify by name when creating and managing storage resources. Policies specify:

- **Storage classes**: Size and performance limits and the types of arrays on which resources may be placed
- **Protection policies**: The types of protection applied to data in allocated resources
- **Host access policies**: The hosts that may access the resources.

Host Access Policy
Storage Class
Protection Policy

Figure 7: Policy objects

Storage Classes

Storage classes define an AZ's storage allocation policies. AZ administrators create storage class templates that specify:

- **Name**: The name by which AZ and tenant space administrators refer to the storage class
- **Hardware_type**: Type(s) of hardware on which resources in the class may be placed (e.g., for FlashArray volumes, FlashArray//XL, FlashArray//X, FlashArray//C)
- **Size_limit, iops_limit, bandwidth_limit**: Maximum capacity and performance limits for resources in the class. For performance, either IOPS, bandwidth, or both may be specified.

To create a storage resource, a tenant space administrator specifies only a storage class name and a size (within the storage class's limit). This ensures that all storage resources managed by Pure Fusion conform to AZ policies.

Data Protection Policies

Providers define data protection policies with which tenant space administrators can specify how the data in their storage resources is to be protected. In the initial Pure Fusion release, data protection policies specify local (i.e., on the array that hosts the resource) snapshot schedules.

Host Access Policies

Tenant space administrators specify the host(s) that may connect to their storage resources by attaching host access policies to them. These policies identify hosts by storage network address, credentials, and a personality (e.g., operating system). For example, a host access policy for iSCSI connections FlashArray volumes, includes one or more parameter sets, each consisting of:

- **iSCSI qualified name (IQN):** The unique name by which an array identifies a host on a TCP/IP network.
- **Challenge handshake authentication protocol (CHAP) credentials**: Credentials used to authenticate host and target before establishing connections.
- **Personality**: Host operating system. Allows Purity to accommodate minor differences in different hosts' iSCSI software implementations.

Host Access Policy Scope

Each host access policy has a scope over which it is recognized:

- **AZ administrator scope**: Allows hosts to connect to any storage resources in the AZ
- **Tenant administrator scope**: Allows hosts to connect to resources in any of the tenant's tenant spaces but not to resources in other tenants' tenant spaces
- **Tenant space administrator scope**: Allows hosts to connect to resources in the tenant space

Figure 8 illustrates the host access policy scope options and circumstances under which each might be used. In all cases, it is the tenant space administrator who attaches host access policies to storage resources. A host access policy can be attached to a storage resource when it is created or later when it is updated.

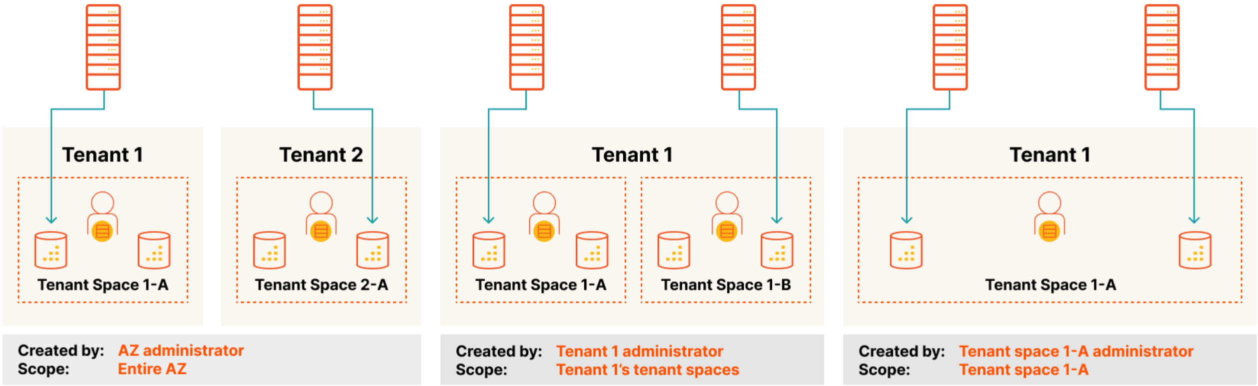

Figure 8: Host access policy scope

Consumer Objects

A major benefit of Pure Fusion is that consumers acquire and manage their own on-premises storage resources without provider involvement. They manage resources via REST APIs that they embed in data center automation tools or in their own user-developed software.

Figure 9 lists the objects through which consumers interact with a Pure Fusion infrastructure.

- Tenants' principal task is managing tenant spaces.
- Tenant spaces are the consumers of storage resources, whose properties their administrators specify by attaching host access and data protection policies to them.

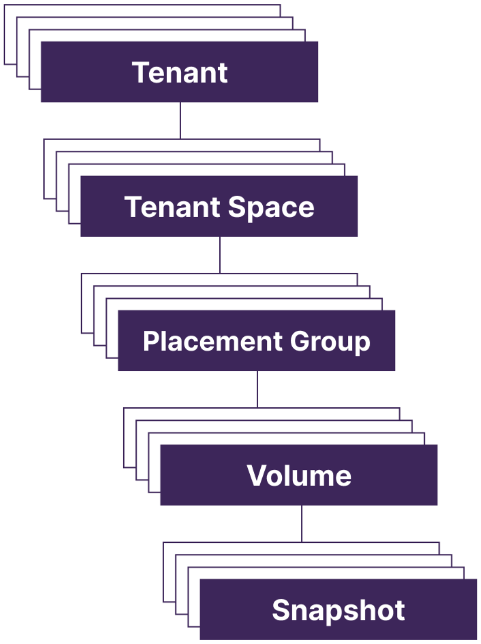

Figure 9: Consumer objects

Virtualizing storage resources allows Pure Fusion to relocate them between arrays within an AZ to rebalance I/O load and/or capacity utilization, to perform maintenance, and to add or replace arrays, all of which are accomplished without consumer impact.

For performance, availability, or similar reasons, however, consumers may require that certain resources be located on the same array, or that they be located separately from others. Pure Fusion provides this capability while still isolating consumers from resources' physical locations through a mechanism called placement groups.

Placement Groups

Placement groups enable tenant space administrators to control the location of storage resources relative to their other resources in the same AZ. Pure Fusion allocates all resources in a placement group on the same array. The resources in a placement group have affinity with each other. When rebalancing, Pure Fusion relocates all resources in a placement group as a unit. Snapshots of resources within a placement group that have the same data protection policy are crash-consistent (i.e., are taken at the same logical instant).

Placement groups also separate resources from other resources, a property sometimes referred to as anti-affinity. Pure Fusion allocates a tenant space's resources assigned to different placement groups on different arrays in an AZ. Similarly, when relocating resources, it ensures that a tenant space's resources in different placement groups are not placed on the same array.

Placement groups are AZ-specific. Figure 10 illustrates two placement groups of volumes in a tenant space. By specifying that data volumes be in Placement group 1 and that backup volumes be in Placement Group 2, the tenant space administrator guarantees that data and backup volumes never occupy the same array, regardless of any rebalancing performed by the provider.

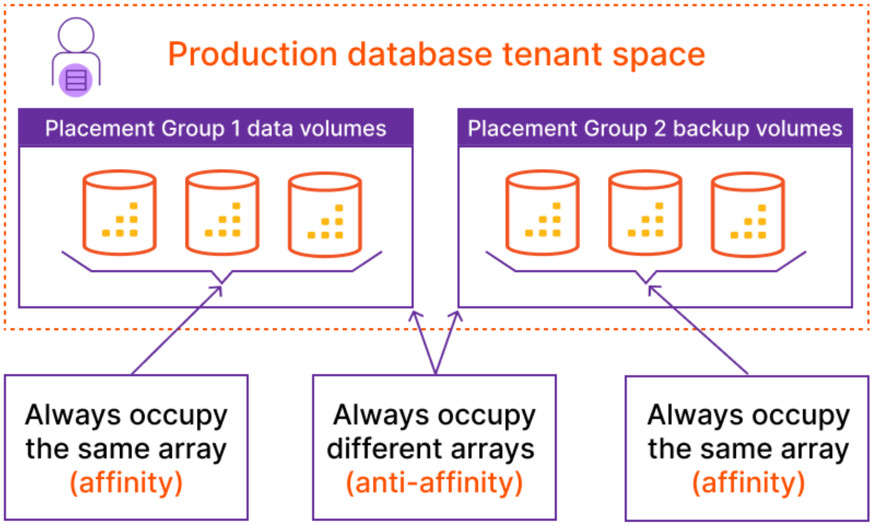
Production database tenant space
Placement Group 1 data volumes
Placement Group 2 backup volumes
Always occupy the same array (affinity)
Always occupy different arrays (anti-affinity)
Always occupy the same array (affinity)

Figure 10: Placement Groups in an AZ

Both tenant and tenant space administrators can create placement group objects. Tenant space administrators can assign storage resources either to their own placement groups or to those created by their parent tenant, but not to those created by other tenants.

Storage Resources

The primary task of tenant space administrators is creation and management of virtual storage resources (e.g., FlashArray volumes). To create a volume, an administrator specifies:

- **Name**: The name by which a tenant space administrator identifies the volume
- **Storage class**: The provider-defined storage class that determines the volume's capacity and performance limits and the type of array on which it is created
- **Size**: The host-visible capacity of the volume (FlashArray volumes are thin-provisioned; regardless of host-visible capacity, they occupy physical space only for data written to them)
- **Placement group**: Affinity with other volumes in the same AZ belonging to the same tenant space and anti-affinity with the tenant space's volumes in other placement groups
- **Protection policy (optional)**: Snapshot frequency and retention times for the volume

Tenant space administrators specify volume sizes, but other properties (performance limits and array types) are determined by the storage class template. Pure Fusion determines the optimal location for each new volume within its specified storage class and placement group constraints; tenant space administrators have no control over the arrays on which their volumes are located.

Other Operations

Tenant space administrators embed Pure Fusion APIs commercial data center automation tools or user-developed software to perform the usual management operations on their storage resources (e.g., resizing, destroying, and eradicating volumes). They control host access to their resources by attaching and removing host access policies, either at creation time or later.

Using Pure Fusion

Figure 11 uses color-coded symbols representing placement groups to show how they might achieve different data availability and application-based replication outcomes.

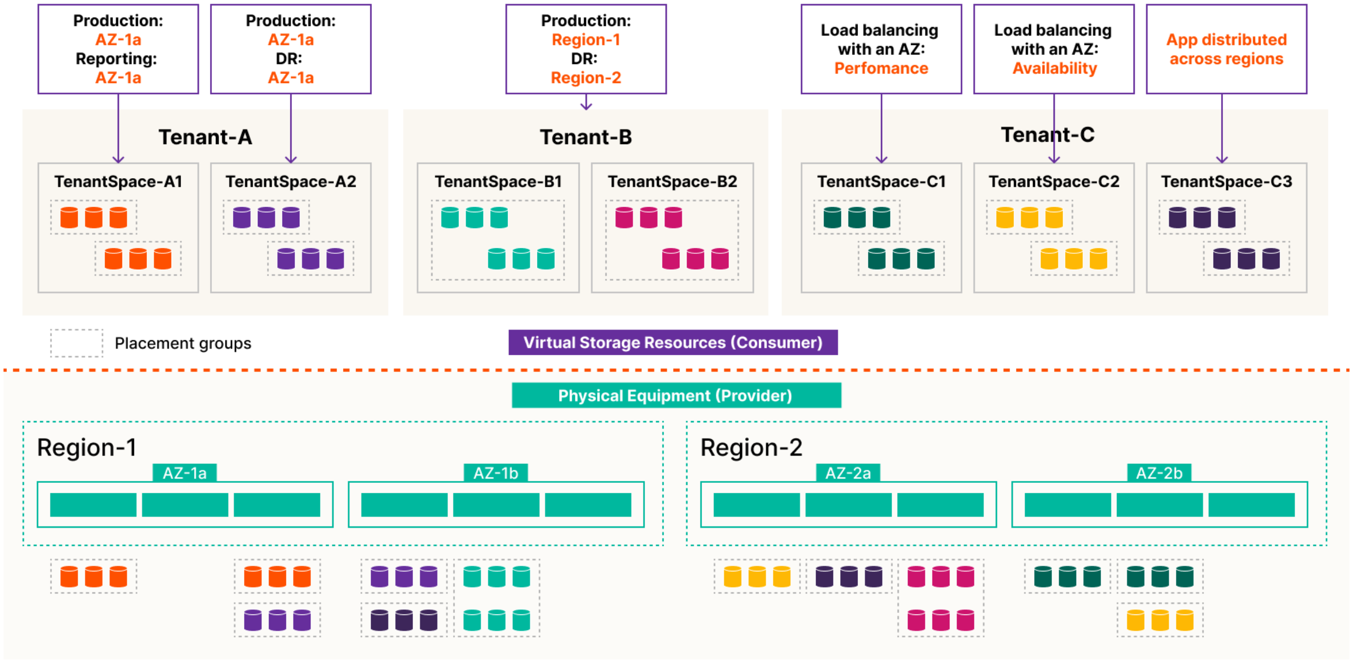
Production: AZ-1a Reporting: AZ-1a
Production: AZ-1a DR: AZ-1a
Production: Region-1 DR: Region-2
Load balancing with an AZ: Perfomance
Load balancing with an AZ: Availability
App distributed across regions
Tenant-A
TenantSpace-A1
TenantSpace-A2
Tenant-B
TenantSpace-B1
TenantSpace-B2
Tenant-C
TenantSpace-C1
TenantSpace-C2
TenantSpace-C3
Placement groups
Virtual Storage Resources (Consumer)
Physical Equipment (Provider)
Region-1
AZ-1a
AZ-1b
Region-2
AZ-2a
AZ-2b

Figure 11: Pure Fusion usage scenarios

The table below describes the six configurations shown in Figure 11 and suggests the outcomes that each might achieve (using host- or application-based replication where applicable).

Graphic	Description	Outcome
🟧🟧🟧	Separate placement groups in the same AZ guarantees that Pure Fusion always locates production and reporting volumes on different arrays	Recoverability, e.g., from host software issues
🟪🟪🟪	Placement groups in different AZs within a region	Local (e.g., within a data center) disaster recovery)
🟩🟩🟩 🟥🟥🟥	Placement groups in AZs in different regions	Remote (e.g., across data centers) disaster recovery)
🟢🟢🟢	Separate placement groups in the same AZ guarantees that Pure Fusion always places different applications' volumes on different arrays	Inter-application load balancing
🟨🟨🟨	Placement groups in different AZs	Enhanced data availability
🟣🟣🟣	Placement groups in AZs in different regions	Support distributed applications

Table 1: Placement group configuration outcomes

Rebalancing

In a world of fast, pervasive networking, ubiquitous flash memory, and an evolving scale-out application architecture, next-gen shared accelerated storage has the power to unite both networked and direct-attached storage in a single, shared architecture. A shared design consolidates data silos, accelerates production, DevOps, and data analytics, and helps enterprises pivot to a data-centric architecture.

Pure Fusion utilizes managed arrays' capabilities (e.g., FlashArray QoS constraints) to distribute I/O load equitably within AZs. It uses the AI-based Pure1 Workload Planner to analyze performance and utilization data from arrays, identify capacity, and/or performance imbalances to determine:

- Optimal placement for new storage resources
- Recommendations for relocating resources to balance utilization or I/O load

When it identifies correctable imbalances, Pure Fusion alerts AZ administrators with recommendations for relocating placement groups. To minimize operational impact, AZ administrators control relocation timing, for example to force relocations to occur outside periods of peak or critical IT activity. Rebalancing does not impact consumer operations.

Figure 12 uses the iSCSI-connected FlashArray volumes supported by the initial Pure Fusion release to illustrate the rebalancing process.

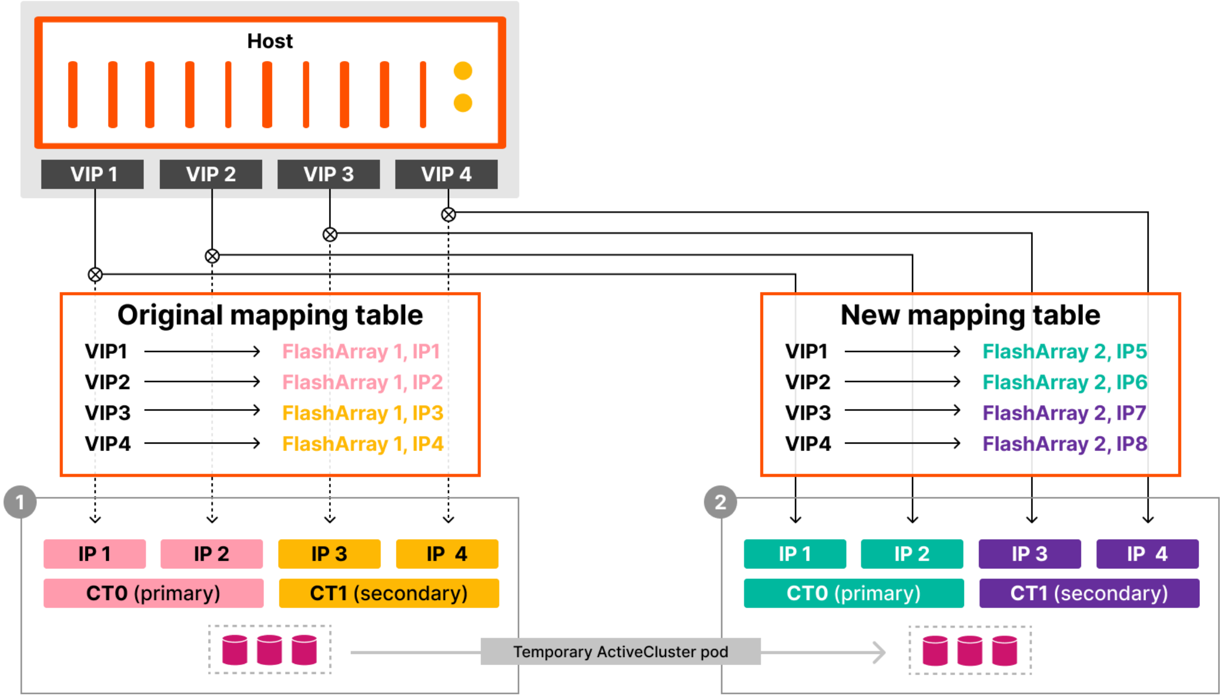

Figure 12: Rebalancing: iSCSI example

When an AZ administrator initiates a rebalance operation, Pure Fusion uses Pure ActiveCluster technology to relocate the placement group's volumes to the target array. During relocation, the VIPs that hosts use to address the volumes are redirected to their new location. When it is safe to do so, the connections to the original array are removed, and the volumes at the original location are destroyed. The entire migration process is transparent to hosts.

Pure Fusion Application Programming Interface (API)

Although provider and consumer CLIs are available, the primary interface to Pure Fusion is through REST API calls embedded in data center automation applications, such as Ansible and Terraform or in user-developed software, rather than through human administrators at consoles. The interface has four key properties. Calls are:

- **Asynchronous**: Calls do not block the caller. They return immediately without waiting for requested actions to complete. Informational calls (e.g., list storage classes) respond with the current persisted infrastructure state; calls that modify state return operation objects that the caller polls to determine the operation's status (Figure 13).
- **Declarative:** Calls express a desired infrastructure state (e.g., from a provider: storage class properties; from a consumer: volume properties). Internally, Pure Fusion takes the actions required to create or modify the object so that the state is attained.
- **Idempotent**: Calls that fail leave the infrastructure in its prior state. There is no need for applications to "roll back" partial modifications, and retrying failed calls is always a proper recovery action.
- **Throughput optimized**: Because Pure Fusion is designed to manage large numbers of arrays programmatically, its design is optimized to provide maximum throughput for many parallel calls rather than the shortest response time for a single call.

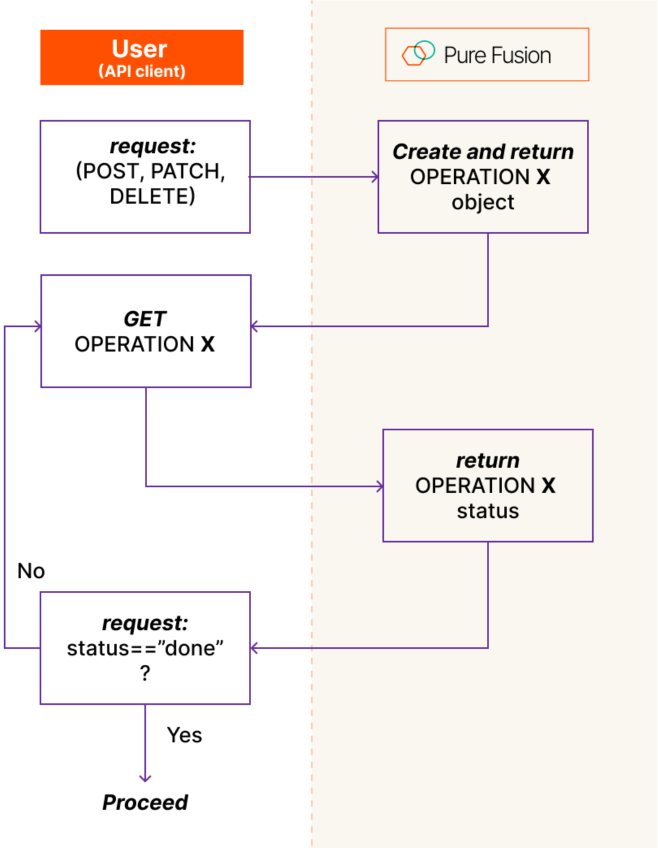

Figure 13: Using Pure Fusion APIs

Invoking APIs

While authenticated Pure1 users with the required access can invoke its services via producer and consumer CLIs, the intent of the Pure Fusion design is software API use. Calls from within software are authenticated by consumer-created objects called API clients. Figure 14 illustrates API client creation and use.

A consumer (typically an application architect or DevOps engineer) generates a public-private keypair (PPK) and communicates the public key to a provider. The provider relays the public key to Pure Fusion, which responds with an API

Client ID. The provider registers a role (administrator or observer) and scope (tenant or tenant space) for the API client with Pure Fusion and sends the API Client ID to the consumer.

The consumer includes the API Client ID in the API calls embedded in its software and digitally signs the calls with its private key generated for the purpose. Pure Fusion authenticates calls using the API Client's public key before issuing them.

An API Client can generate additional API Clients without human intervention. For example, a tenant administrator might use this capability to create an API client with tenant scope that would create an API client with tenant space scope for each of its tenant spaces. This would allow tenant space administrators to run software to manage only their own resources.

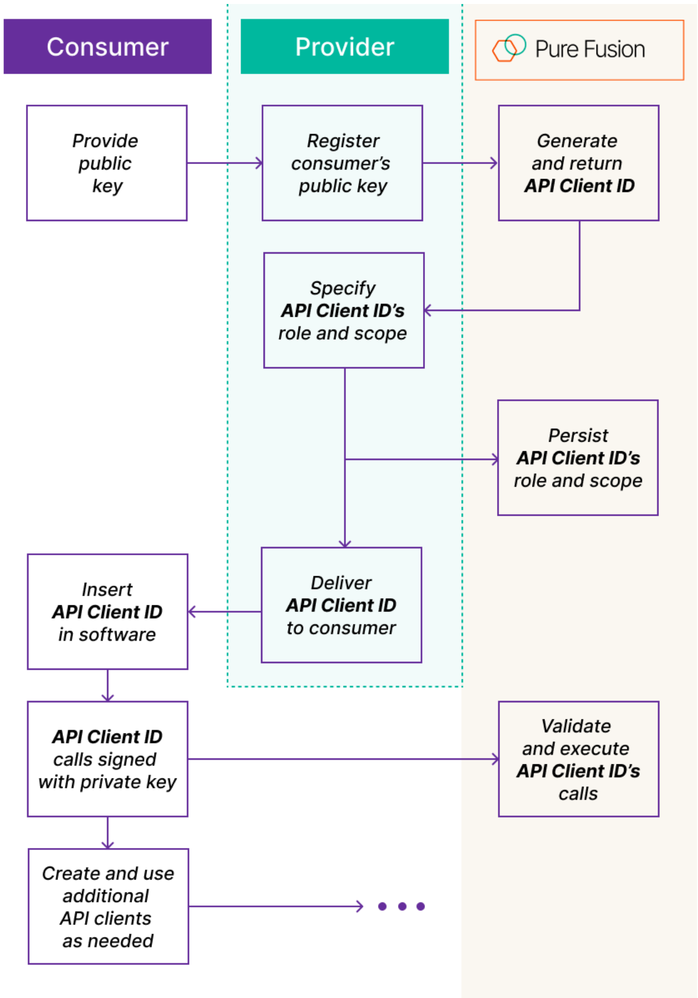
Consumer
Provider
Pure Fusion
Provide public key
Register consumer's public key
Generate and return API Client ID
Specify API Client ID's role and scope
Persist API Client ID's role and scope
Insert API Client ID in software
Deliver API Client ID to consumer
API Client ID calls signed with private key
Validate and execute API Client ID's calls
Create and use additional API clients as needed
• • •

Figure 14: API clients

Getting Started with Pure Fusion

With Pure Fusion, IT teams typically fulfill the storage provider role, consolidating management of selected arrays and making self-service storage available to consumers. To begin using Pure Fusion, an organization's IT team would:

- Coordinate with Pure Storage field representatives and support engineers to enable organization-specific Pure1 cloud-based management
- Define and publish storage class templates, data protection policies, and host access policies that specify the types of resources available to consumers, how data in them can be protected, and what hosts may have access to them
- Direct array administrators to enable edge management on selected arrays and install Pure Fusion agent software on them
- Create regions and AZs and assign managed arrays to them
- Create tenant objects for their organization's major users of storage.

Tenants would then create tenant spaces for their applications with tenant space administrator accounts to enable self-management of storage resources.

The foregoing steps establish the Pure Fusion infrastructure that makes self-service storage management possible. Tenant space administrators can then create placement groups and storage resources.

Pure Fusion Rollout

As it has with other innovations throughout its history, Pure is introducing Pure Fusion in stages, starting by meeting the most urgent customer requirements and adding capabilities over time. As initially released, Pure Fusion supports volumes on iSCSI-connected FlashArrays in multiple AZs and regions, with intra-AZ placement group relocation voluntarily initiated by providers in response to advisory alerts from Pure1.

The company expects to analyze other potential capabilities (e.g., support for FlashArray File Services, Cloud Block Store (CBS), FlashBlade, additional storage network technologies, replication, etc.) and services (e.g., automatic relocation) to determine that they are (a) non-disruptive and (b) uniformly beneficial to users. Based on analysis and demand, additional capabilities may be added to Pure Fusion over time.

Conclusion

Pure Fusion changes storage management in several ways, but the net result is cloud-like self-management of on-premises storage that scales to petabytes. With Pure Fusion, consumers manage their own resources (within policy constraints) without concern for their location. Providers manage arrays, storage classes, data protection policies, and tenants to optimize results and provide organization-wide consistency with negligible application disruption.

The key innovations that make this possible are separation of producer and consumer roles, policy objects, and an API-centric design. Separating the storage services from the infrastructure that provides them, and enabling management by software, make it possible to automate the management of on-premises storage to a greater extent than has previously been possible, while providing a better experience for:

- **Providers**: Intra-organization multi-tenancy becomes possible. Infrastructure-wide monitoring and analysis provide the information needed to utilize physical storage capacity efficiently by placing and relocating storage resources without impacting consumers' operations.
- **Consumers**: Self-service, on-premises storage becomes a reality, guaranteeing consistency with organizational storage access and management policies. With automation, consumers' storage resource requests can be satisfied in minutes rather than in days or weeks.

The Pure Storage products and programs described in this documentation are distributed under a license agreement restricting the use, copying, distribution, and decompilation/reverse engineering of the products. No part of this documentation may be reproduced in any form by any means without prior written authorization from Pure Storage, Inc. and its licensors, if any. Pure Storage may make improvements and/or changes in the Pure Storage products and/or the programs described in this documentation at any time without notice.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. PURE STORAGE SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Pure Storage, Inc.
650 Castro Street, #400
Mountain View, CA 94041