

# Technical Disclosure Commons

---

Defensive Publications Series

---

September 2021

## Correlation Tracing in a Distribute Storage System

Grant Griffiths  
*Pure Storage, Inc.*

Luis Pabón  
*Pure Storage, Inc.*

Follow this and additional works at: [https://www.tdcommons.org/dpubs\\_series](https://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Griffiths, Grant and Pabón, Luis, "Correlation Tracing in a Distribute Storage System", Technical Disclosure Commons, (September 13, 2021)

[https://www.tdcommons.org/dpubs\\_series/4595](https://www.tdcommons.org/dpubs_series/4595)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.



# PURE STORAGE DEFENSIVE PUBLICATION

## Correlation Tracing in a Distribute Storage System

Grant Griffiths

Luis Pabón

## Motivation

---

Large storage systems distributed across different data centers, racks, nodes, and components can be very hard to debug when a problem arises. With different container orchestrators and possible ingress paths, problematic API requests quickly become impossible to trace within these systems. With the rise of microservices architectures combined with monolithic storage systems, tracing a single request from a container orchestrator to a low-level storage element can be like finding a needle in a haystack.

## Description

---

A package of context information, called a Correlation Context (e.g., a “context” object), can be generated and used to trace where and when a request originated through logging. Each and every log emitted by a storage system may log this correlation context, making debugging a single request through a multi-faceted storage system far easier to debug. The Correlation Context may be implemented in a container system, such as in a container orchestrator (e.g., Kubernetes) and/or in a storage platform used by the container orchestrator.

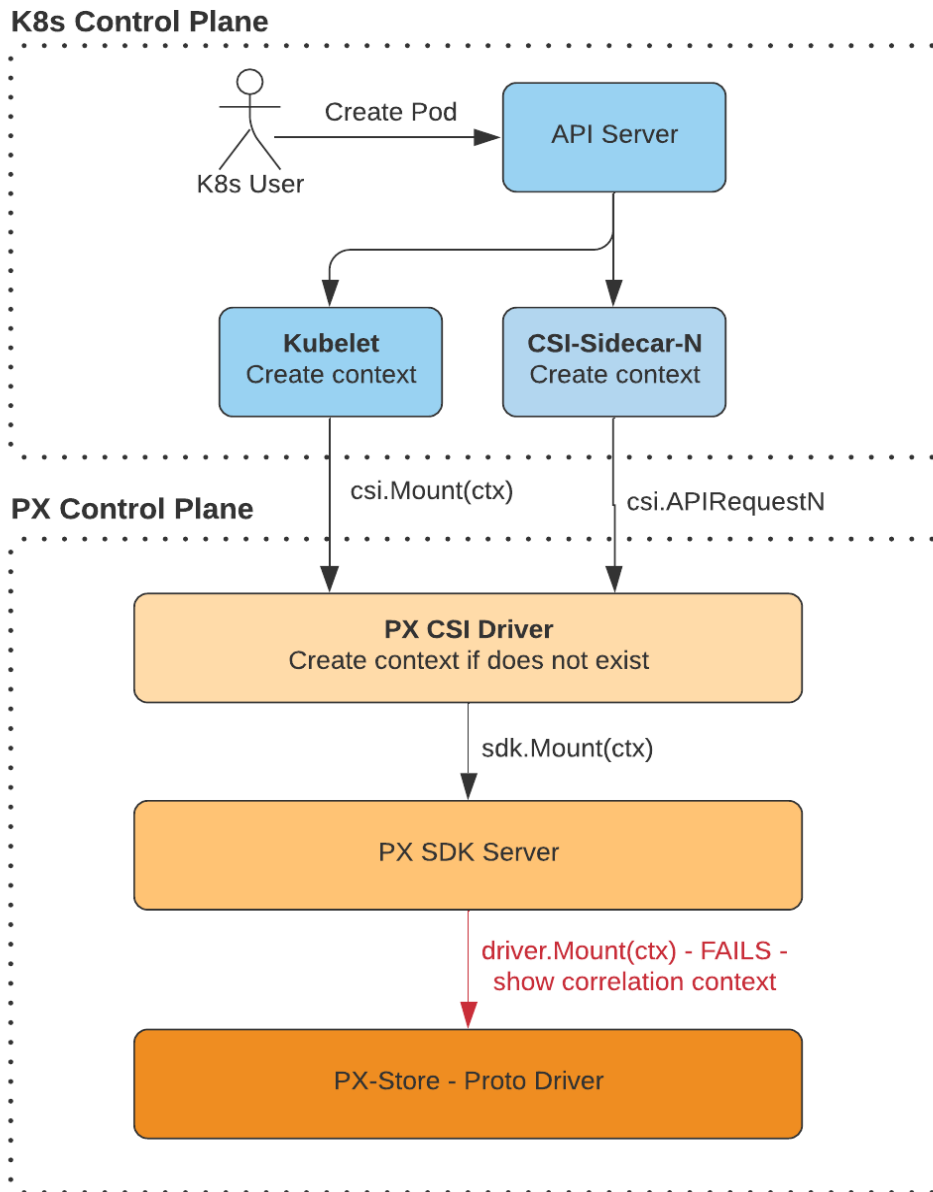
## Use Cases

---

- Given a container orchestration system that supports correlation context:
  - When a CSI driver receives a request,
    - Should be able to see the request origin, start time, correlation UIID
    - Should add any relevant correlation information.
  - When a problem arises related to a request:
    - Should be able to see a package of correlation information for performing root cause analysis
    - Should see the correlation ID in any alerts or events passed to the user.
- Given a container orchestration system that does not support correlation context:
  - When a CSI request reaches the storage system,
    - Should generate a correlation context
- Given have received a bug report and an end user has received a correlation ID in their error:
  - When performing an RCA,
    - Should be able to find the problematic request and any associated logs via the correlation ID
    - Should be able to find any remote requests forwarded to other nodes in the cluster.

## Design

The following design depicts the above scenario where a Kubernetes user creates a pod. This pod works through all components - Kubernetes control plane, Portworx (PX) CSI Driver, PX SDK Server, and finally almost reaches the PX-Store (proto driver) layer. This final call fails and will log the correlation context.

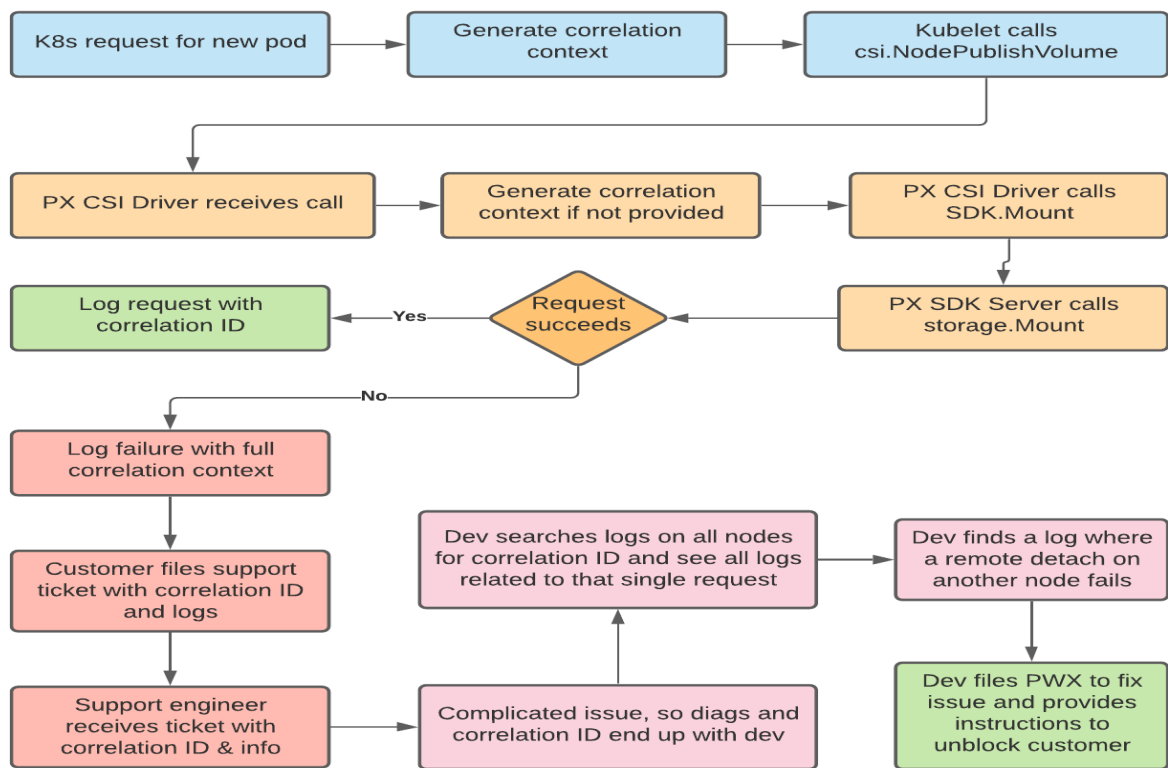


## Solution

1. Generate a "context" object to pass around the system
  - o **Correlation-ID:** Randomly generated ID
  - o **Origin:** How this request entered the system
2. Log this context object in the system
  - o Failures, info logs, etc.
3. Include correlation-id in error messages and events
  - o Support engineers can use this to perform root cause analysis

## Use case: Customer support flowchart

The following flowchart shows how a request may flow through the various PX Components and may aid in a customer support ticket.

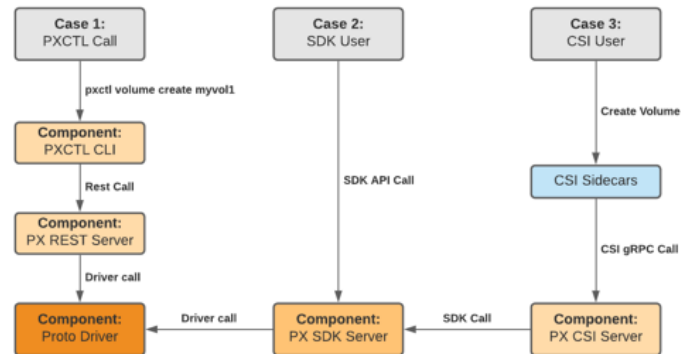


## Importance of Component Tracking

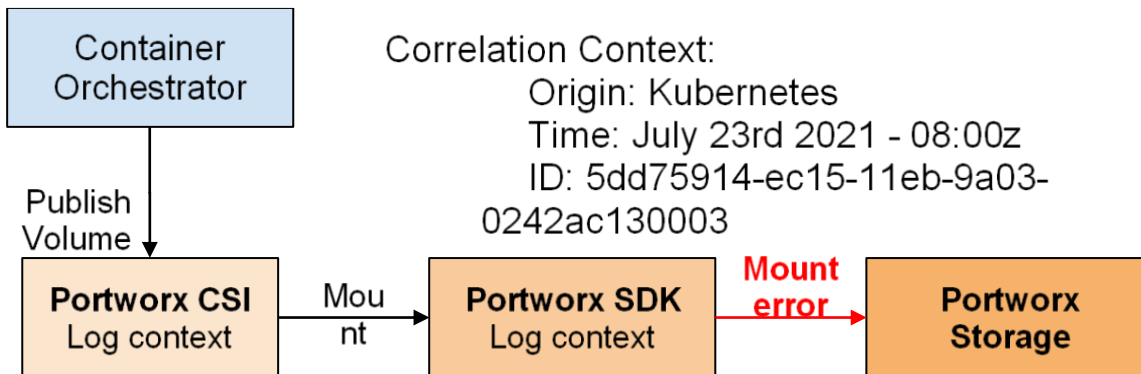
1. SDK receives a sdk.MountVolume API request.  
 Generate correlation context with the following values:
  - a. Correlation-id: 123
  - b. Origin: csi-driver
  - c. Component: csi-driver
2. Log line emitted in SDK Server:
 

```
msg="SDK volume unmount"
component=sdk-server correlation-id=123 origin=sdk-server
```
1. SDK Server request to proto.CreateVolume, passing correlation context
2. Log line emitted in Proto Driver:
 

```
msg="MOUNT FAILURE - Mountpath not mounted"
component=proto-driver correlation-id=123 origin=csi-driver
```
3. Return error to end-user with correlation-id=123



## Solution Detail



### Where:

- **Container orchestrator:** A container scheduler such as Kubernetes, Nomad, or DCOS
- **Portworx CSI:** A CSI driver that works with any container scheduler and relays calls to the Portworx SDK Server
- **Mount error:** In this case, an error occurred when the SDK server was calling the Portworx storage mount call. We will log all correlation context when this failure occurs, so that we can accurately identify where and when this call originated.

## Work breakdown

---

- Add standard logging package that includes correlation context
- Pass context from CSI driver down all the way to proto driver level
- Replace existing logs with standard logging pkg, starting w/ CSI driver layer

## About the Authors

---



Grant Griffiths is a member of the technical staff at Pure Storage. Grant is interested in the cloud native ecosystem, is an active Kubernetes and Kubernetes-CSI contributor, and was a speaker at #GopherconUK 2018 and #kubcon EU 2020. He holds a dual bachelor-of-science degree in computer science and mathematics from Syracuse University.



Luis Pabón is a member of the technical staff at Pure Storage. Luis has been an architect and technical leader for an array of software applications and frameworks at a number of engineering organizations. He holds degrees in Electrical and Computer Engineering from the University of Massachusetts Amherst and Worcester Polytechnic Institute.





Pure Storage, Inc.  
Twitter: [@purestorage](https://twitter.com/purestorage)  
[www.purestorage.com](http://www.purestorage.com)

650 Castro Street, Suite #260  
Mountain View, CA 94041

T: 650-290-6088  
F: 650-625-9667

Sales: [sales@purestorage.com](mailto:sales@purestorage.com)  
Support: [support@purestorage.com](mailto:support@purestorage.com)  
Media: [pr@purestorage.com](mailto:pr@purestorage.com)  
General: [info@purestorage.com](mailto:info@purestorage.com)