

The I's Have It

Identity, Immutability, Idempotence, and Interchangeability Form the Backbone of Distributed Computing

Pat Helland

Salesforce (PHelland@Salesforce.com)

This article is a subset of a paper called "Identity by Any Other Name" to be published soon in ACM Queue.

ABSTRACT

When constructing a scalable system, there comes a time when you can't hold everything in a single relational database. When that happens, data and computation are knit together using various types of identifiers. This paper examines the essential role of identifiers in the implementation of three other essential properties, immutability, idempotence, and interchangeability. Really, these four "I's are how it all works!

INTRODUCTION

Distributed systems are usually implemented with identifiers which may be called by many names including IDs, names, keys, numbers, URLs, file names, reference, UPCs (Universal Product Codes), ISBNs (International Standard Book Number) and more. Scalable homogeneous as well as scalable heterogeneous systems use these identifiers to connect the data, computation, and long-running work together. The use of identifiers empowers the fundamental properties of distribution!

IDENTITY AND IDEMPOTENCE

Idempotence is the property that says it's OK to do work more than once. If it happens *at least once*, the behavior is the same as if it happens *exactly once*.^[1] In general, idempotence is a subjective concept that ignores side effects outside of the plane of abstraction provided by the service.^[3]

Idempotence frequently depends on having an identity for the work. You need to understand the identity of the operation to decide if you've done it before. There are other cases such as reading a record where the work is naturally idempotent because it leaves no visible effects as it's performed. If changes are made, tracking that it's already done requires identity of some kind.

Sometimes, the identity used to provide idempotence is a consequence of some connection or session. That works until a new session arrives to retry the failed session.

Managing the requester's identity, the target's identity, and the identity of the work in question are some of the hardest problems in scalable systems needing idempotence.

As they clear checks, banks use a simple approach to identify them and to provide idempotence. They use two basic tricks:

- The transaction's identity is the preassigned check number.
- The check must clear in less than one year after written.

The second constraint limits the list of cleared checks the bank must maintain while preserving exactly once processing.

This article is published under a Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), which permits distribution and reproduction in any medium as well as allowing derivative works, provided that you attribute the original work to the author(s) and CIDR 2019.

9th Biennial Conference on Innovative Data Systems Research (CIDR '19) January 13-16, 2019, Asilomar, California, USA.

IDENTITY AND IMMUTABILITY

Immutability is the property that something doesn't change. No matter how many times the data is read, the same result is returned. Immutability is the basis for many of today's solutions, from low-level hardware to massively scalable solutions.^[2]

Immutability is a relationship between an identity and an unchanging result.

The semantics of the "unchanging" result may be slippery. Just as side-effects in idempotent operations are ignored, equivalent results for the identity may be interpreted as immutable answers.

Amazon reports over 7,000 results for "King James Bible" each with identical text and many with a unique and different ISBNs!

Immutability is in the eye of the beholder and identity captures the ambiguity of the beholder's interpretation of "unchanging".

IDENTITY AND INTERCHANGEABILITY

Interchangeability can be viewed as a duality with immutability. Rather than asking, "Is this thing identical?" to what we had before, we ask, "Is this thing equivalent?". *Is it good enough?*

When manufactured items are all brand new and identical, you're fine accepting any one of them from the warehouse, assuming they're not damaged. There is an identity for the product, and that identity means any one of them will do. *They are interchangeable.*

When reserving a room at a hotel, you accept that one king-sized nonsmoking room is as good as another—even if one is next to the elevator and really noisy. ***There is an identity for each individual room and another identity for one of the equivalent pool.*** You reserve one from the pool of rooms without knowing exactly which one.

An identifier for a product description in a product catalog refers to an ambiguous version of the product description. That's OK... any one will do, as the versions are interchangeable.

Identities may refer to an abstract grouping of equivalent things.

The judicious use of ambiguity and interchangeability lubricates distributed, long-running, scalable, and heterogeneous systems.

The real art of interchangeability lies in finding a way to identify the equivalent set of individuals...

CONCLUSION

Idempotence, immutability, and interchangeability are deeply related to identity. They are the fulcrum of distributed systems.

REFERENCES

- [1] Helland, P. 2012. Idempotence is not a medical condition. *acmqueue* 10(4), 30; <https://dl.acm.org/citation.cfm?id=2187821>.
- [2] Helland, P. 2016. Immutability changes everything. *acmqueue* 13(9); <https://queue.acm.org/detail.cfm?id=2884038>. (First printed in CIDR 2015).
- [3] Helland, P. 2017. Side effects, front and center! *Communications of the ACM* 60(7), 36-39; <https://dl.acm.org/citation.cfm?id=3080010>.