Achieving Digital Permanence

RAYMOND BLUM WITH BETSY BEYER

THE MANY CHALLENGES TO MAINTAINING STORED INFORMATION AND WAYS TO OVERCOME THEM

Digital permanence has become a prevalent issue in society. This article focuses on the forces behind it and some of the techniques to achieve a desired state in which "what you read is what was written." While techniques that can be imposed as layers above basic data stores—blockchains, for example—are valid approaches to achieving a system's information assurance guarantees, this article won't discuss them.

First, let's define *digital permanence* and the more basic concept of *data integrity*.

Data integrity is the maintenance of the accuracy and consistency of stored information. *Accuracy* means that the data is stored as the set of values that were intended. *Consistency* means that these stored values remain the same over time—they do not unintentionally waver or morph as time passes.

Digital permanence refers to the techniques used to anticipate and then meet the expected lifetime of data stored in digital media. Digital permanence not only considers data integrity, but also targets guarantees of relevance and accessibility: the ability to recall stored data and to recall it with predicted latency and at a rate acceptable to the applications that require that information.

To illustrate the aspects of relevance and accessibility, consider two counterexamples: journals that were safely stored redundantly on Zip drives or punch cards may as well not exist if the hardware required to read the media into a current computing system isn't available. Nor is it very useful to have receipts and ledgers stored on a tape medium that will take eight days to read in when you need the information for an audit on Thursday.

THE MULTIPLE FACETS OF DIGITAL PERMANENCE

Human memory is the most subjective record imaginable. Common adages and clichés such as "He said, she said," "IIRC (If I remember correctly)," and "You might recall" recognize the truth of memories—that they are based only on fragments of the one-time subjective perception of any objective state of affairs. What's more, research indicates that people alter their memories over time. Over the years, as the need to provide a common ground for actions based on past transactions arises, so does the need for an objective record of fact—an independent "true" past. These records must be both immutable to a reasonable degree and durable. Media such as clay tablets, parchment, photographic prints, and microfiche became popular because they satisfied the "write once, read many" requirement of society's record keepers.

Information storage in the digital age has evolved to fit the scale of access (frequent) and volume (high) by moving to storage media that record and deliver information in an almost intangible state. Such media have distinct advantages: electrical impulses and the polarity of magnetized ferric compounds can be moved around at great speed and density. These media, unfortunately, also score higher in another measure: fragility. Paper and clay can survive large amounts of neglect and punishment, but a stray electromagnetic discharge or microscopic rupture can render a digital library inaccessible or unrecognizable.

It stands to reason that storing permanent records in some immutable and indestructible medium would be ideal—something that, once altered to encode information, could never be altered again, either by an overwrite or destruction. Experience shows that such ideals are rarely realized; with enough force and will, the hardest stone can be broken and the most permanent markings defaced.

In considering and ensuring digital permanence, you want to guard against two different failures: the destruction of the storage medium, and a loss of the integrity or "truthfulness" of the records.

Once you accept that no ideal medium exists, you can guard against both of these failures through redundancy. You can make a number of copies and isolate them in different failure domains so some of them can be counted on to survive any foreseeable disaster. With sufficient copies kept under observation through frequent audits and comparison, you can rely on a quorum of those copies to detect and protect the record from accidental or deliberate alteration.

Copies made for these purposes have different motivating factors, which can be placed into two categories:

- Backups – Copies that protect operations from failures caused by an act of nature or neglect.
- Archives – Copies made to preserve the record from the forces of change, be they deliberate or accidental.

INFORMATION PERMANENCE IN THE DIGITAL AGE

Before the 1970s disembodied information did not exist outside of gossip or bardic lore. For thousands of years, knowledge was preserved by altering physical artifacts: from the 3000 BCE rations of Mesopotamian beer to the 1952 tax rolls of the state of Rhode Island, giving permanent life to a fact meant marking a clay tablet, parchment scroll, or paper punch card. Setting aside the question of its truth, the fact of record existed in plain sight, made permanent in chiseled marks or insoluble ink for the life of the artifact. While fire, flood, or fugitive dye might have challenged the durability of the records, barring destruction or theft, it was reasonable to assume that the artifacts of record would remain consistent. A date of birth or tax payment committed to the official record would be the same when recalled for the next decade's census or audit.

In the post-Renaissance and post-Industrial Revolution eras, as humanity embarked upon more and more endeavors with time spans of decades or years, the amount of information that was critical for society to retain exploded. Typesetting and printing processes were optimized and automated to scale up with the increasing need for recorded information. While codices and microfiche use space far more efficiently than clay tablets or scrolls, society cannot dedicate an infinite amount of space to storing copies of birth records and articles of incorporation.

Then came the Information Age.

Suddenly, it seems that nothing can be allowed to slip into obscurity: street maps, bank account records, personal timelines, birthday party videos—all are recorded and stored. While they may lie unused for decades or centuries, we fully expect that the data will be available for later research or perusal. As the volume of historical records surges, the classic model of devoted storage artifacts—be they stone, paper, or plastic—cannot keep up, a perfect manifestation of the adage, “It doesn’t scale.” Paper is too bulky and takes too long to write on and to read. It’s safe to say that recording the history of the world in tangible, physical form is no longer feasible.

Conveniently and not coincidentally, the very same technological advances that created this problem of too much information also led to available solutions. We now have the ability to store information in a “purer” electronic form, broadly and commonly referred to as digital media. The electronic digital representation of information is accomplished with far less energy and space than with older physical or “analog” recording techniques. To use a very coarse measure for comparison: whereas a typical book might weigh 12 ounces and contain 80,000 words, the same amount of information can be stored as 3.2 million bits, which occupies 1/10,000th of a commonly available micro SD (secure digital) card that weighs 0.016 ounces. Compared to a paper novel, that SD card has at least 7.5 million times the information per ounce (and this ignores the application of various techniques to increase the efficiency of digital storage space, such as compression and de-duping).

As the density of information has increased, recording and reading rates have necessarily increased by a similar order of magnitude. If you record 100,000 times more information, but do so at the same rate of transcription, you will accumulate quite a backlog of facts, figures, and news articles to be committed to permanent records. Luckily, it takes far less energy and time to flip the state of submicroscopic bits than to carve notches in stone or to drag a pen to deposit ink on a sheet of paper.

Not surprisingly, while these faster, more fluid storage media are a blessing in one aspect, they are a curse in another. State that is easily set is also easily unset, either unintentionally or maliciously. RAM, flash memory, and magnetic disks can be corrupted through chance interactions that are far more lightweight than actions that can wipe out older, physical media. It might take an intense, persistent building fire to destroy file cabinets full of marriage certificates in the basement of a hall of records, but some stray electromagnetic emission could wipe out the same information stored on a couple of SSDs (solid-state drives). To make matters worse, it's immediately obvious that your basement was on fire, but you might not know that the contents of your SSDs were corrupted until months or decades later when you need to access the data they once contained.

Concerns over the permanence of recorded information were easily addressed in the past—mechanisms such as stone, archival-quality papers and inks, and fireproof vaults provided well-understood and easily implemented assurances that records would survive for predictable periods of time. The lifetime of encoding techniques was rarely an issue unless you encountered records made in an obsolete language (such as the Egyptian hieroglyphs that modern people couldn't decipher until the Rosetta Stone decree was discovered).

Permanence has become a very real problem as storage techniques and media churn rapidly. While you can rely on a medium such as stone or parchment for a historically demonstrated value of permanence, the impermanence of modern media such as magnetic tape, CD-ROMs, and flash memory has been a surprise. The evolution of paper production in the mid-19th century perhaps foreshadowed this trend. As demand for mass-printed material increased, printers shifted from rag-based paper to more quickly and cheaply produced lignin-rich wood-pulp paper. As a result, archivists and comic-book collectors were surprised and disappointed by the fragility of the cheaper medium.

If paper was a disappointment, at least its permanence faced no challenges beyond the durability of the medium itself: reading a page requires only the sense of sight, which hasn't changed much since the earliest written records were made. Digital media have introduced new concerns: you cannot directly sense the information on, for example, a flash-memory module; you need specialized equipment to interpret the impressions left on digital storage media, and this equipment must be available and able to provide an interface relevant to current information-processing systems. In short, having well-preserved magnetic tape isn't enough: you need a functioning tape drive, and you must be able to interface the tape drive with your computing system.

In addition to being less permanence-resilient than older, nonelectronic (hereafter, *analog*) storage media, digitally stored data is subject to yet another pressure: the increasing demands for precision in this data-driven world requires unerring reliability. While a measure of 10 acres or three pounds would have been accepted with some understood or even expected margin of error in the past, today's expectations are increasingly precise: 13 ounces or 310 euros must mean exactly that. The world demands both a growing amount of relevant and necessary data and better "quality" or precision of that data.

Not coincidentally, these demands align with the shift from analog values and media to their digital counterparts: a drawn line may be perceived as crossing the Y axis at "just around" 10, but a recorded digital metric is either 10 or it is not. When using a slide rule, precision is tied to the perception and visual acuity of the operator, whereas an electronic calculator displays a precise, viewer-agnostic value out to many digits of precision. Modern society also expects immediate results: queries should be answered in real time and transactions should complete almost immediately, so that dependent actions can proceed.

The overall effect of this set of forces is simply summarized: we need to store ever more information (greater breadth), of higher precision or resolution (greater depth), while maintaining or decreasing the latency of access (greater throughput). The increased relevance of the information (greater impact) to people's lives demands higher fidelity from storage techniques (greater reliability). We need digital permanence.

CATEGORIZING FAILURE MODES

Any number of triggers can introduce failure modes of storage techniques and media, but there are some broad categories of failure to help identify the most likely vulnerabilities and effective means of mitigation:

- Latent failure incurred by the passage of time – Staleness of media, bitrot.
- Failure introduced by *force majeure* events – Typically site disasters such as earthquake, fire, flood, electromagnetic pulse, or asteroid impact.
- Failures caused by malevolence or ignorance – Most often, exploitations of process deficiencies.
- Failures caused by usage in unanticipated operation sequence or volume – Usually planning deficiencies.
- Failures resulting from flaws in systems or their components – Bugs and a lack of isolation or a way to contain their effects.

Failures also have distinct timelines or life cycles:

- Big bang – Significant amounts of data are affected at once. An event or atomic operation causes systemic harm.
- Slow and steady – Corruption or loss trickles into a data store at a rate that is probably on the same order of magnitude as normal access, perhaps as a side effect of normal operations.

The scope of a failure can also be classified:

- Widespread – Large, broad swaths of data are affected, seemingly without regard for discriminators within the data.

TABLE 1: **FAILURE CATEGORIZATION MATRIX**

CATEGORY	TIMELINE	SCOPE
Introduced over / by time	Big bang	Widespread
Force majeure		
Malevolence or ignorance		
Unanticipated usage	Slow and steady	Narrow and directed
Defects		

- Narrow and directed – Specific subsets of the stored data are affected, presumably with some discernable pattern that a domain expert would recognize.

A given failure will have at least one value for each of these three aspects—category, timeline, and scope—so the potential failures can be visualized as a matrix, shown in table 1.

According to this matrix, a comprehensive view of risk should take up to 20 (5×2×2) different failure modes into account. An effective plan for gauging and ensuring digital permanence within this system must include either a means to mitigate each of these possible failure modes, or acknowledgment of unaddressed risks. The likelihood and impact of each failure must also be quantified in some way. No matter how comprehensively (or superficially) you plan on handling a given failure, you should recognize what it is and how much it may cost you. This analysis will help prioritize your budget for ensuring digital permanence and disaster-recovery planning.

MITIGATING RISKS TO DIGITAL PERMANENCE

These failure modes are as similar as chocolate and concrete (apples and oranges actually *do* have a lot in common). It follows that appropriate mitigations are also wide ranging. While keeping a full offline data store copy is a reasonable failsafe for a big bang (for example, widespread loss caused by an asteroid slamming into a data center), this tactic isn't ideal for guarding against user error that deletes one account's transactions for the past business day. Your response to this diversity of risks might be to diversify your platforms, avoiding failure caused by a vulnerability specific to one platform. Defense via platform diversity has its appeal but also its drawbacks—stitching together myriad and diverse media, transfer rates, and vendor support levels can become an overwhelming task in itself, leaving little time for your day job.

The complexity of this problem space calls for a well-reasoned strategy for achieving digital permanence in a given system. This section examines methods for codifying coverage of two different aspects of digital permanence in a system, broadly categorized as *data integrity* and *accessibility*.

Preserving data integrity

The data integrity goal is fairly easily stated: If you store some value V in a system, indexed or identified as K, you expect to be able to call up K at some later time and be certain that the value retrieved is, in fact, V. The inherent problem here is one of *trustworthiness*: the system should be relied upon to do its job. If the retrieved value was in fact $V2 \neq V$, how would you know? If your application is expected to constantly checksum and verify the storage layer's operations, you're experiencing a major abstraction leak and are almost certainly on your way to writing a spectacular *God class*.

A better strategy is to implement a set of guarantees and checks outside of any client application—operations that are conceptually part of the storage system(s). These operations aim to detect and recover from the failures that a storage system may encounter, independent of any current or future client system. Table 1 discusses failure modes and means to address them somewhat generically; specific implementations will be defined for and by the system under scrutiny.

The examples in table 2 are not meant to be exhaustive; rather, they provide a sufficiently large example to

TABLE 2: **EXAMPLE SET OF FAILURES AND THEIR MITIGATIONS**

FAILURE MODE	MEANS TO MITIGATE
a. Force majeure x big bang x widespread	1. Standby failover serving site 2. Remote data store mirror
b. Introduced over / by time x slow and steady x narrow and directed	3. Parameterized snapshot restore and manual adjustments
c. Introduced over / by time x big bang x widespread	4. Re-create data store from log replay
d. Defects x slow and steady x narrow and directed	3. Parameterized snapshot restore and manual adjustments
e. Force majeure x big bang x narrow and directed	1. Standby failover storage site 2. Remote data store mirror 3. Parameterized snapshot restore and manual adjustments 4. Re-create data store from log replay
Malevolence or ignorance x big bang x widespread	4. Re-create data store from log replay

illustrate the recommended methodology. Column 1 identifies a set of failure modes, and column 2 provides mitigations for each failure mode. The numerals in column 2 identify overlap in the pool of processes and mechanisms so that you can optimize the ROI for each technique used. The goal is to obtain the most coverage for the smallest investment.

To make the best use of this table, you need to be able to weigh the different failures and mitigations so you can prioritize solutions. Table 3 rates the impact of each failure mode.

Table 4 shows the relative cost of fully implementing

TABLE 3: **IMPACT OF EACH FAILURE MODE**

FAILURE MODE	IMPACT
a. Force majeure x big bang x widespread	Catastrophic
b. Introduced over/by time x slow and steady x narrow and directed	Medium
c. Introduced over/by time x big bang x widespread	Catastrophic
d. Defects x slow and steady x narrow and directed	Medium
e. Force majeure x big bang x narrow and directed	Low
f. Malevolence or ignorance x big bang x widespread	Catastrophic

TABLE 4: **COST OF EACH MITIGATION**

MEANS TO MITIGATE	COST
1. Standby failover serving site	High
2. Remote data store mirror	Medium
3. Parameterized snapshot restore and manual adjustments	Low
4. Re-create data store from log replay	Medium

each proposed mitigation technique.

Now that you understand the options, their relative costs, and their relative values, you can optimize to find the best coverage per cost. The final set of mitigation techniques is optimized with the following parameters:

- All failure modes with an impact other than low must be addressed, but you should provide mitigation techniques for all failure modes if there's no additional cost.
- The lowest-cost option to mitigate a given failure mode is preferred.
- A mitigation technique, once implemented, is applicable to all failure modes for which it is effective.
- Implement as few mitigation techniques as possible in order to minimize the operational complexity of the system.

Table 5 combines the data from tables 2, 3, and 4. The data in table 5 can be sliced to reveal both the mitigations that provide the broadest coverage and the lowest-cost mitigation for each failure. Note that column [e] is considered optional because failure modes of this category typically have relatively low impact. It's a welcome bonus if you can cover column [e] by piggybacking on mitigations already being implemented for other failure modes.

Broad coverage. Consider a complete data integrity plan to include any set of rows (mitigations) from table 5 that together provide a value in every column (failure modes). For example, by implementing mitigations in rows [2], [3], and [4], you can achieve complete coverage because each failure mode (column) is addressed.

Lowest-cost mitigation. In addition to coverage, you should consider the total cost of a set of mitigations. For

TABLE 5: **COST VS. COVERAGE OF MITIGATION TECHNIQUES**

FAILURE X MITIGATION	a. Force majeure x big bang x widespread	b. Introduced over / by time x slow and steady x narrow and directed	c. Introduced over/by time x big bang x widespread	d. Defects x slow and steady x narrow and directed	e. Force majeure x big bang x narrow and directed	f. Malevolence or ignorance x big bang x widespread
1. Standby failover serving site	High				High	
2. Remote data store mirror	Medium				Medium	
3. Parameterized snapshot restore and manual adjustments		Low		Low	Low	
4. Re-create data store from log replay			Medium		Medium	Medium

example, the relative costs of rows [1] and [2] might lead you to exclude row [1], as it has a higher cost and provides no additional coverage. If you were optimizing to mitigate failure mode [e], you would choose [3], the lowest-cost applicable mitigation technique.

This exercise does not take into account the likelihood of given failure modes. This factor is highly variable based on the specific failure of a given category: for example, "Asteroid Impact" as an instance of failure mode [a] or "Bad Software Release" as an instance of [d]. The specific failures that a system may experience and their likelihood are dependent on the details of the system being evaluated. When planning data integrity for a given system, prioritizing relevant work, and allocating resources, the individual failures in each category and their likelihood should be enumerated and averaged or summed to account for the likelihood of failure.

Now that a framework has been established for preserving the integrity of data stores, let's turn to a second aspect of digital permanence, called *relevance* or *accessibility*.

Maintaining accessibility

No matter how securely you've locked away hermetically sealed copies of your information, placing every conceivable safeguard in place, there are two surprisingly common snafus that cause the best-laid plans to go awry:

- You can no longer read the data in its preserved form.
- Restoring the data is too expensive to be feasible.

The first issue, one of obsolescence, is well illustrated by an example already given: the ancient Egyptians placed great importance on the fidelity of religious texts and recorded them in stone—the most permanent information storage available. They failed to anticipate that their chosen encoding scheme, hieroglyphs, would be obsolete by the fourth century CE. As a result, their information, although preserved with high integrity, would be as good as gone for millennia, indecipherable until a translation function in the form of the Rosetta Stone was recognized in 1799. Closer to home, consider the family photos stored on several Zip disks along with a spare Zip drive and EISA (Extended Industry Standard Architecture) card in a fireproof box. While this was a seemingly thorough archive strategy in 1995, it wasn't thorough enough to make those photos readily accessible using 2018 technology.

You can most simply keep all means of access for all of your data relevant through exercise: employ full end-to-end tests or rotate the live or shadow service through different data stores to validate them. Do it often enough to provide time to address a deprecated storage medium or strained network route before they become totally inaccessible or useless.

The second issue that affects accessibility is one of scale. Somewhat obviously, the more information you have to process in a given operation, the more resources the processing will take. While transactions are written one at a time, perhaps resulting in a few kilobytes of information per storage operation, restoring a snapshot of data accumulated over months could result in a single storage "operation" from the storage user's point of view—a restore that has to process terabytes or petabytes of information.

That doesn't come cheap. At transfer rates of common buses such as USB 3.0, the *theoretical* minimum transfer time for a petabyte of data is close to 56 hours. If you're restoring your customer-facing online service's data, you're not likely to have the luxury of more than two days of unavailability.

At some point, you will have to exploit the classic tradeoff of space vs. time, designing parallelism into your data integrity processes to make sure that the information remains accessible and relevant within acceptable time thresholds. While you may not be able to escape the worst-case scenario of needing to transfer that petabyte, you could perform that transfer with 100 concurrent workers, reducing the 56 hours to less than an hour of wall time, saving your users and your business.

Of course, this strategy is easier said than provisioned. Ultimately, you need to examine the total cost of recovery vs. cost to your business to find the sweet spot. It's a good idea to model a range of scenarios to guide you in determining the resources to devote to data integrity operations. This process is well modeled in a spreadsheet. To return to the previous example: at one end of the spectrum you model the cost of 100 provisioned workers plus the total cost to the business of a one-hour outage; the other end of the spectrum includes the relatively low cost of one provisioned worker plus the presumably high cost to the business of a 56-hour outage. You should include intermediate points such as 10 workers and an outage of close to six hours in the analysis to help find the optimal parameters of your provisioning, communications plans, and playbooks.

Defense in depth

"When it rains it pours," "Trouble comes in threes," "Le disgrazie non vengono mai sole": there's no shortage of idioms that warn against taking a breather from threats to digital permanence. These threats never go away. There are myriad ways for this pessimistic prediction to manifest. Multiple failures in the failsafe are a common and especially capricious twist of fate: just as you breathe a sigh of relief in the middle of a disaster recovery because you've diligently backed up your data to tape, the tape breaks in the drive. Or you might experience a perfect storm of failures: a network outage causes intermittent timeouts of write operations for users accessing application servers in western Europe, while at the same time, the system that stores transaction logs goes offline when a blue heron flies into an open transformer panel at a data center.

Roll your eyes and laugh now, but what can happen will happen, so your plans should employ the principle of defense in depth to protect your systems from compound or overlapping failures. Remember that these points of failure don't know about each other and are as likely to happen concurrently as they are to happen at different times.

Bitrot: The forces of decay and neglect

Obsolescence of some critical function or component of a mitigation plan is the most common root cause of disaster-recovery failures. When you've worked hard to come up with a plan to address an unpleasant, annoying, or even painful issue, it's natural and reasonable to want to put it out of your head and punt follow-up from your calendar. Unfortunately, it's dangerous to do so. Any system in motion is changing and evolving, so it's important to respond with accordingly flexible plans. If your plans don't match the elasticity of the situations they're meant to deal with, the mismatch will lead to decreasing relevance of the plan as the system diverges ever further from its former state.

Bitrot can manifest in many ways: access-control lists expire, resource reservations become obsolete or unavailable, or playbooks are unfamiliar to new staff. There is one simply stated guideline to detect and counter bitrot: practice, practice, practice.

Practicing your recovery plans

A backup shouldn't be taken for granted or viewed as an end goal: try restoring from it; replay transaction logs periodically; failover between alternate sites. These are the operations that you should care about, so make sure that they still actually work as designed. Perform mitigation exercises with a frequency determined by the failures that they address. For example, failover between sites is used to mitigate big bang failures, and therefore should be performed on a noncontinuous basis, perhaps weekly or monthly. Log replay is used to recover from steady-state failures. Therefore, more frequent, continuous, or O(days) end-to-end tests of this operation are appropriate.

In addition to establishing how often to exercise data integrity operations to ensure your expected digital permanence, you need to define the proper scope of these test exercises. The closer your exercise is to a full end-to-end operation, the greater your confidence in it will be. For a failover between alternate sites, consider actually switching among alternate sites regularly, rather than viewing one site as primary and others as failover or backup sites. Running log recovery against test accounts or regularly selected sets of accounts will either assure you that log replay is currently a trustworthy operation or point out its shortcomings so you can fix any problems or at least know not to rely on this strategy in the event of a failure.

MAKING IT LAST AND KEEPING IT TRUE

Every era has introduced new societal challenges when developing and dealing with technological advances. In the Industrial Age, machining methods evolved to produce more, better, and previously undreamt of machines and tools. Today's Information Age is creating new uses for and new ways to steward the data that the world depends on. The world is moving away from familiar, physical artifacts to new means of representation that are closer to information in its essence.

Since we can no longer rely on the nature of a medium to bestow permanence, we must devise mechanisms to do so that are as fluid and agile as the media to which we're entrusting our information and ever-increasing aspects of our lives. We need processes to ensure both the integrity and accessibility of knowledge in order to guarantee that history will be known and true.

Related articles

➡ How Do I Model State?
Let Me Count the Ways
A study of the technology and sociology of Web services specifications
Ian Foster, et al.
https://queue.acm.org/detail.cfm?id=1516638

➡ Keeping Bits Safe: How Hard Can It Be?
As storage systems grow larger and larger, protecting their data for long-term storage is becoming more and more challenging.
David S. H. Rosenthal
https://queue.acm.org/detail.cfm?id=1866298

➡ META II: Digital Vellum in the Digital Scriptorium
Revisiting Schorre's 1962 compiler-compiler
Dave Long
https://queue.acm.org/detail.cfm?id=2724586

Raymond Blum *leads an engineering team in Google's Developer Infrastructure that is charged with keeping thousands of Google engineers productive and happy. He was previously a site reliability engineer at Google, for which he was somewhat prepared by running a hosting services company. Blum spent previous lives developing software for media and financial companies. In what spare time exists, he makes friends with robots.*

Betsy Beyer *is a technical writer for Google Site Reliability Engineering in New York City and the editor of* Site Reliability Engineering: How Google Runs Production Systems *and The* Site Reliability Workbook. *She has previously written documentation for Google data centers and hardware-operations teams. She holds degrees from Stanford and Tulane.*