Census Security Whitepaper

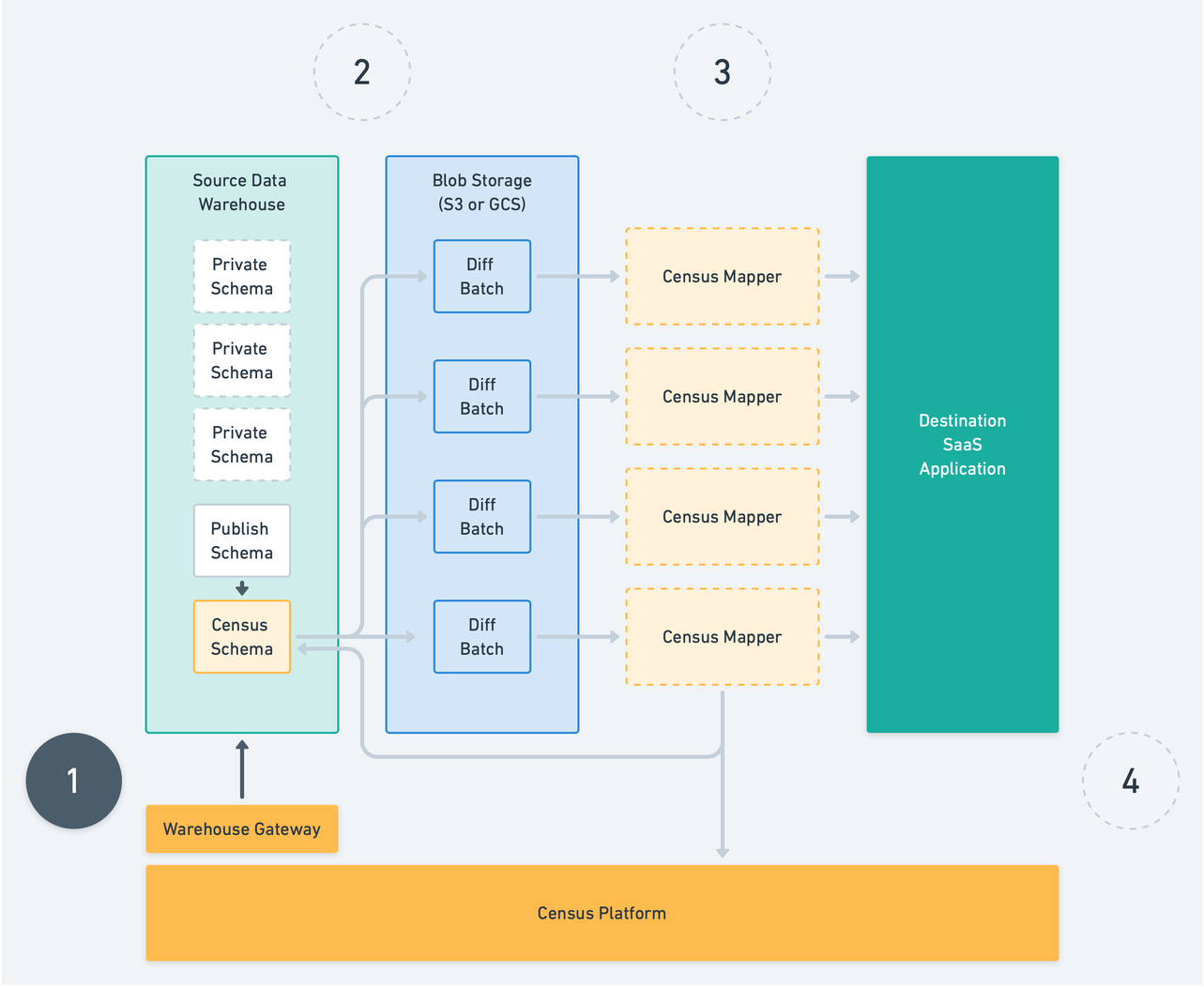
A New Approach to Customer Data

Unlike other data platforms you may have used, the Census synchronization and publishing pipeline is designed around the core principle of storing and handling as little of your customers' data as possible. Using the powerful capabilities of cloud data warehouses and SaaS applications, Census is able to perform most of the "logic" for determining what records need to be synced and how to match those records to your existing data within your own warehouse. When your customer data is handled by Census' applications servers, we use ephemeral workers that do not store your data, and practice data defense-in-depth with multiple layers of data scrubbers (managed by our platform and our cloud computing partners) to ensure that none of your sensitive data is mistakenly left behind on our servers or in our cloud storage.

This whitepaper describes how Census syncs work, in which cases Census will handle or store your sensitive data, and the various techniques the Census platform employs to minimize the risk of a data breach. Our architecture is always evolving to support higher performance publishing to more destinations, but our commitment to handling your data securely remains our north star, and we believe the best way to ask for you to trust us with your data is to handle it as little as possible.

How it works

Step 1: Identify Changes in Warehouse

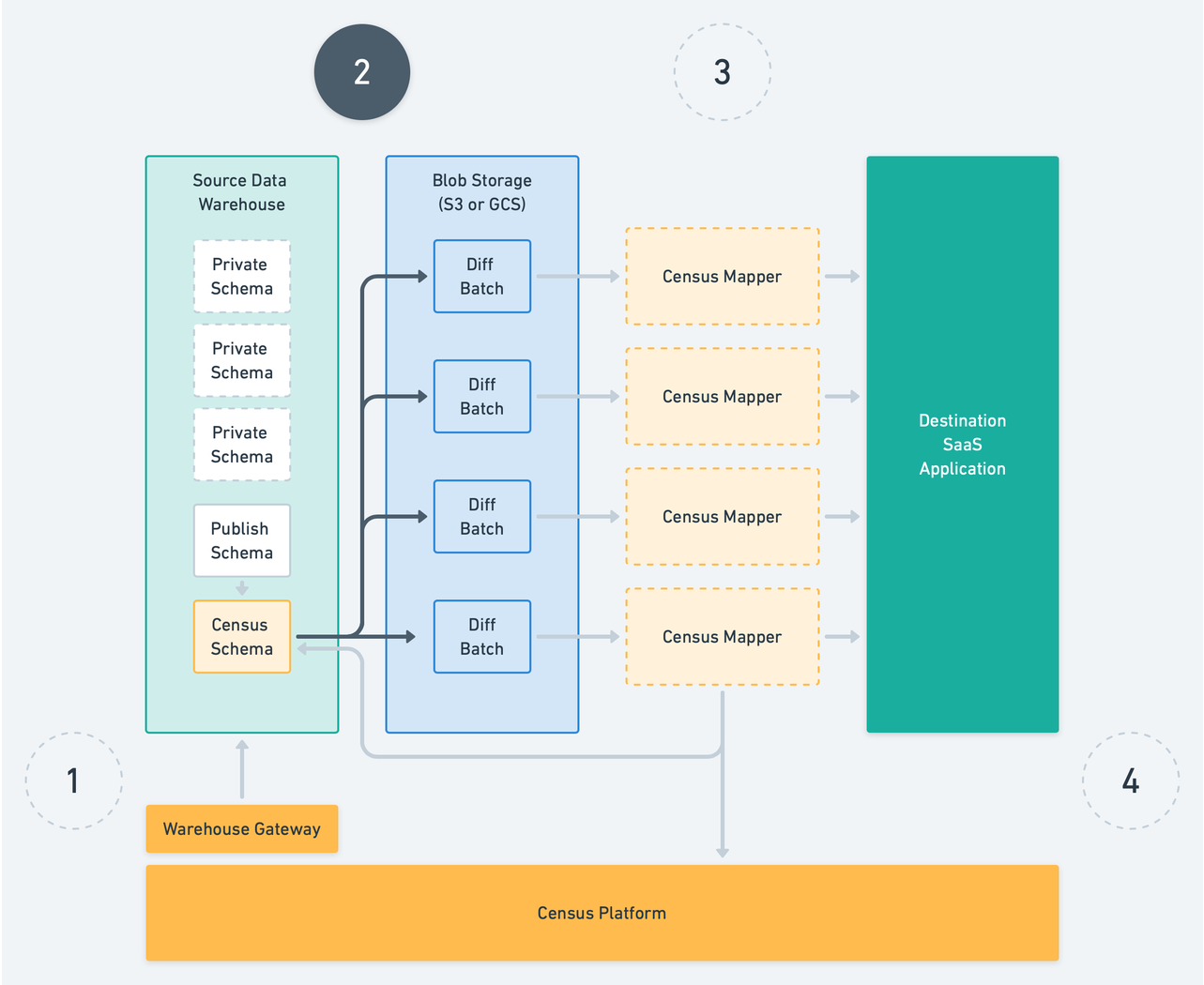


As part of onboarding, you'll create a "Census schema" within your data warehouse. This is a bookkeeping area used by the platform to keep track of what changes have occurred to your data and what still needs to be synced. Instead of storing this data in our platform, we keep it in your data warehouse for two reasons. The first is improved security: in the event of a breach of the Census platform, your customer data is not at risk because we do not store it. The second is efficiency: we can save load on your warehouse and provide a more cost-effective platform by avoiding unnecessary copying back-and-forth between your systems and ours.

Census will access your data warehouse through a dedicated gateway, which always uses an encrypted connection and originates from a set of well-known static IP addresses that you can add to the allowlist for your warehouse for additional control. The Census user account that accesses your warehouse is not a "superuser" - it is a least-privilege account that can only read from the schemas that you choose and

can only write to the Census bookkeeping schema. It's impossible for Census to read data from your warehouse that you don't explicitly opt-in, and if you choose you can even set up fine-grained table-level access for the Census user.
When a new sync starts, Census looks for rows of data that have been added or changed since the last sync, and records a temporary snapshot of those rows in the Census schema. Your warehouse reports back to the Census platform only the number of rows that have changed or been added - the actual data is not copied to Census servers.

Step 2: Unload "Diffs" to Cloud Storage



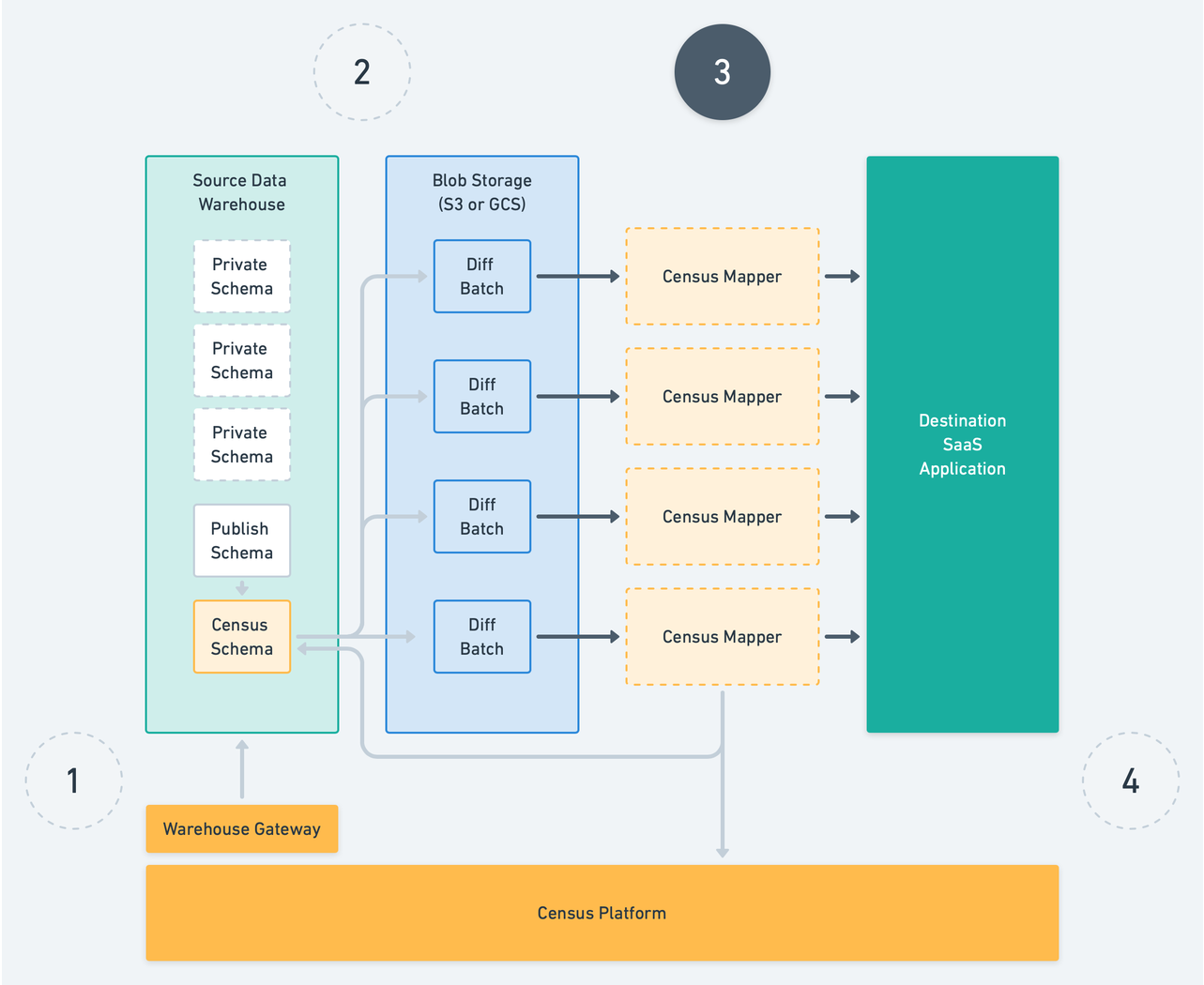
Once the data differences, or "diffs", have been calculated, Census instructs your warehouse to copy just those rows to our cloud provider's "blob storage" bucket (either AWS S3 or Google Cloud Storage). The temporary credentials we provide to your warehouse that are used to copy data are only capable of writing data, not

reading it back out, so this is a one-way data flow. Diffs are assigned cryptographically unique key paths in the cloud storage bucket, making it impossible for an attacker to guess paths to customer data.

The cloud storage buckets used by Census are configured with two additional security measures, managed by AWS and Google Cloud.

First, items in these buckets are automatically removed after seven days. This means that even if the Census platform crashes or goes offline, Amazon or Google will ensure that in the worst case your data is only left in the temporary blob storage for one week. In most cases this interval is much lower - Census proactively removes this temporary data as soon as it is processed in step 3 - but this removal rule serves as a useful insurance policy. Second, data in these buckets is encrypted using the cloud providers' server-side encryption, protecting your data from some classes of attacks on the cloud providers themselves.

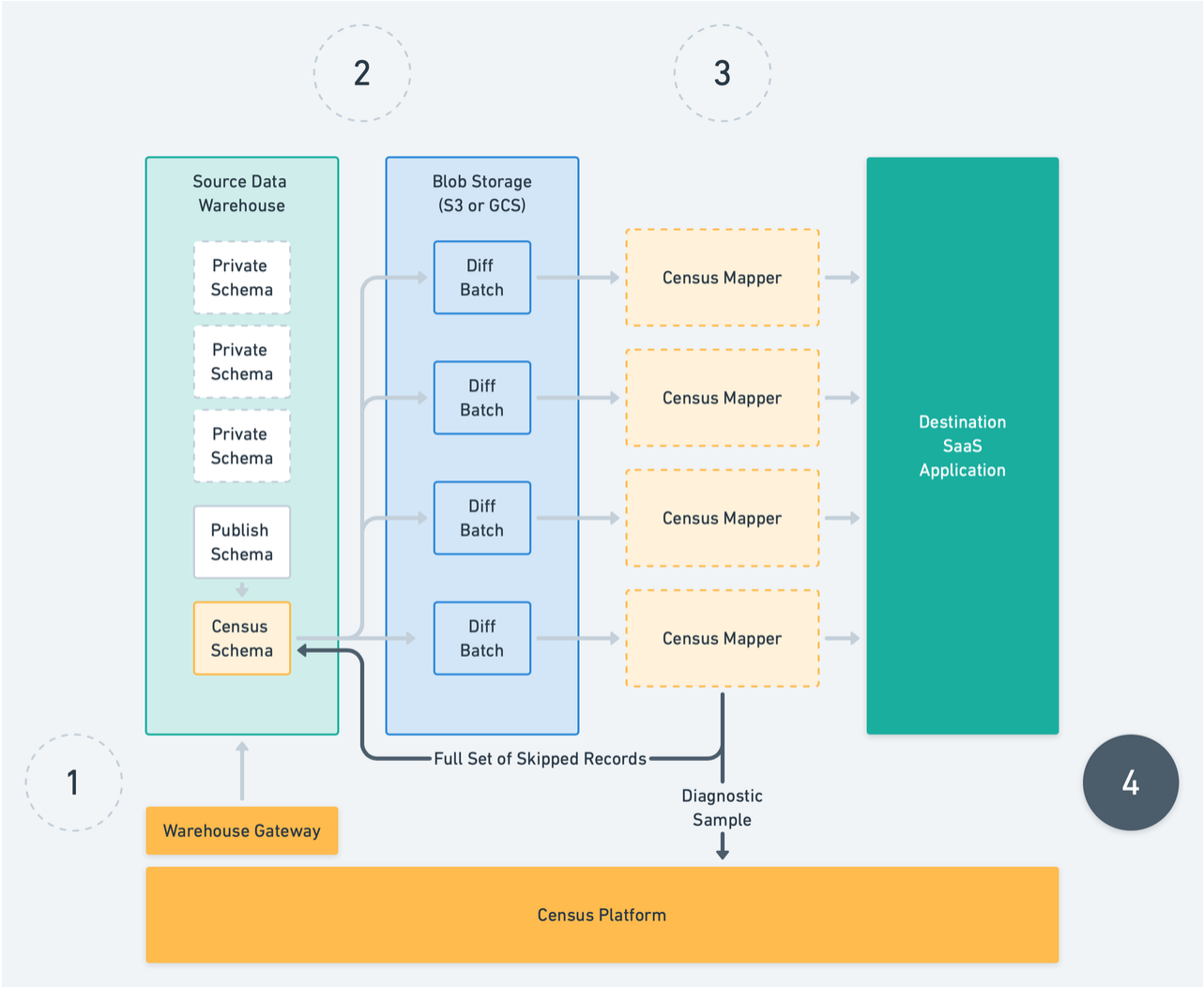
Step 3: Prepare Data for Loading into SaaS Applications



In most cases the data from the warehouse cannot be loaded directly to the SaaS application without changes. Census needs to adjust data types and formats, apply your mapping rules to translate columns in the warehouse to fields and attributes in the SaaS app, check to see if the data already exists in the SaaS in order to decide whether to create or update data, and handle any errors or data validation issues. Instead of bringing the data into the core Census platform, Census starts up a fleet of stateless, ephemeral mappers (similar to “Lambdas” or “Cloud Functions”) that each pull in one batch of diffs and apply it to the service.

Mappers use separate temporary credentials that are only capable of reading data from blob storage for least-privilege isolation. Once a mapper finishes its work, it deletes the diff batch from cloud storage and reports its results (see step 4).

Step 4: Report Skipped Records and Feedback to Warehouse



When a mapper finishes its work, it has two lists of records - those that were successfully loaded to the SaaS application, and those that failed, either because of validation issues in the data or transient errors (SaaS outages or errors, networking issues, etc). The list of records that failed are written directly back to the Census schema in your data warehouse, and a small sample of those failures (no more than 100) are captured and sent to the Census platform. These diagnostic samples are the only customer data that will ever be stored in Census, and their storage is limited to 7 days.