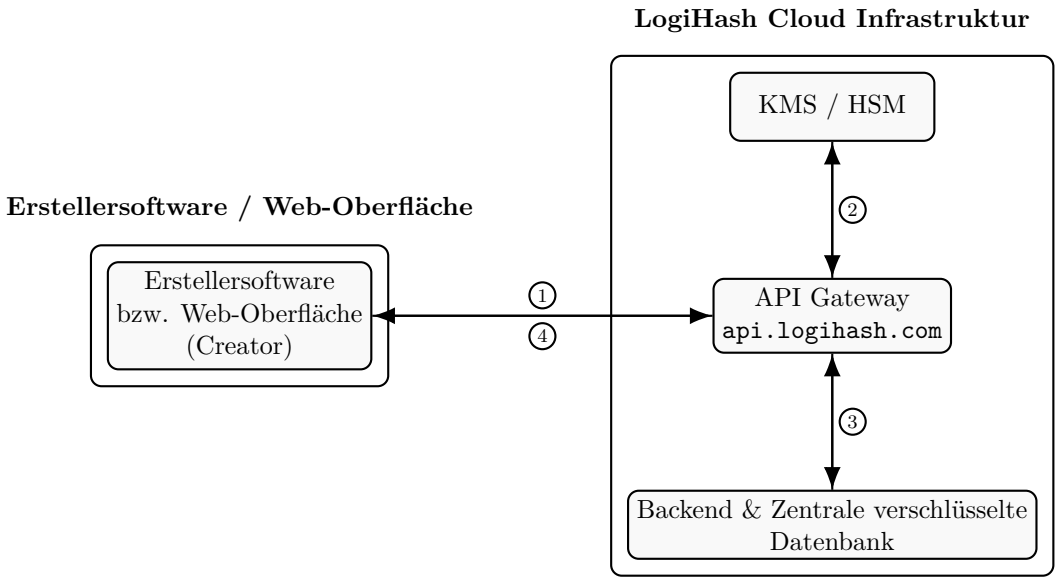
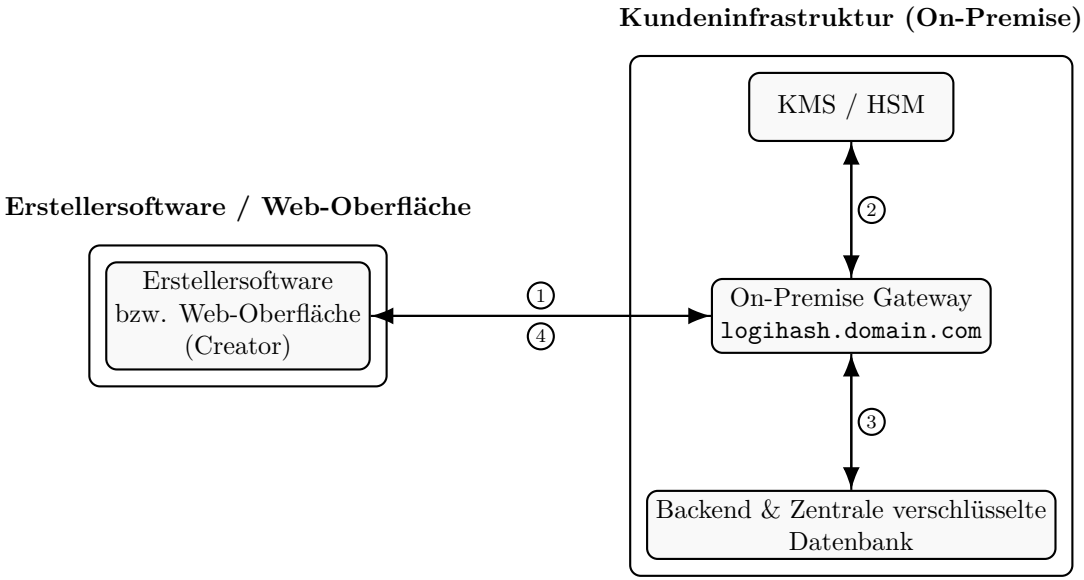
1. Cloud-basierte LogiHash-Lösung – T-Code-Erstellung (Generierung)



Prozessbeschreibung – Cloud-basierte T-Code-Erstellung (Schritte 1–4):

1. **Daten senden:** Die Erstellersoftware bzw. Web-Oberfläche sendet Dokumentdaten (Creator, Adresse, Dokumenttyp, Zeitstempel, optionale Felder, Darstellungsoptionen) an `api.logihash.com` (z. B. `POST /api/v1/tcode/generate`).
2. **Schlüssel:** Das API-Gateway fordert bzw. verwendet Schlüssel vom KMS/HSM.
3. **Verschlüsseln, Signieren, Speichern:** Die Dokumentdaten werden verschlüsselt, die T-Code-Bausteine werden erzeugt und signiert, und die verschlüsselten Nutzdatensätze werden in der zentralen Datenbank gespeichert.
4. **Antwort:** Das Gateway liefert die T-Code-Zeichenkette und ein PNG-Bild des T-Codes an die Erstellersoftware bzw. Web-Oberfläche zurück.

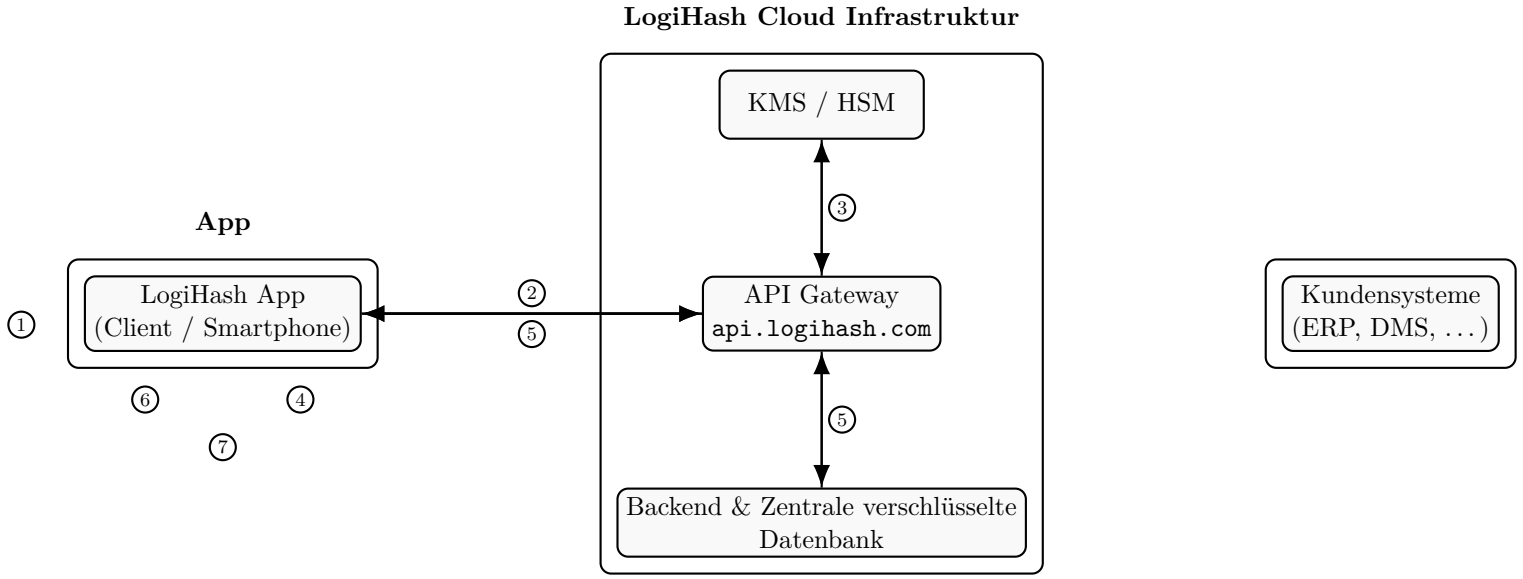
2. On-Premise LogiHash-Lösung – T-Code-Erstellung (Generierung)



Prozessbeschreibung – On-Premise T-Code-Erstellung (Schritte 1–4):

1. **Daten senden:** Die Erstellersoftware bzw. Web-Oberfläche sendet Dokumentdaten (Creator, Adresse, Dokumenttyp, Zeitstempel, optionale Felder, Darstellungsoptionen) an `logihash.domain.com` (z. B. `POST /api/v1/tcode/generate`).
2. **Schlüssel:** Das API-Gateway fordert bzw. verwendet Schlüssel vom KMS/HSM.
3. **Verschlüsseln, Signieren, Speichern:** Die Dokumentdaten werden verschlüsselt, die T-Code-Bausteine werden erzeugt und signiert, und die verschlüsselten Nutzdatensätze werden in der zentralen Datenbank gespeichert.
4. **Antwort:** Das Gateway liefert die T-Code-Zeichenkette und ein PNG-Bild des T-Codes an die Erstellersoftware bzw. Web-Oberfläche zurück.

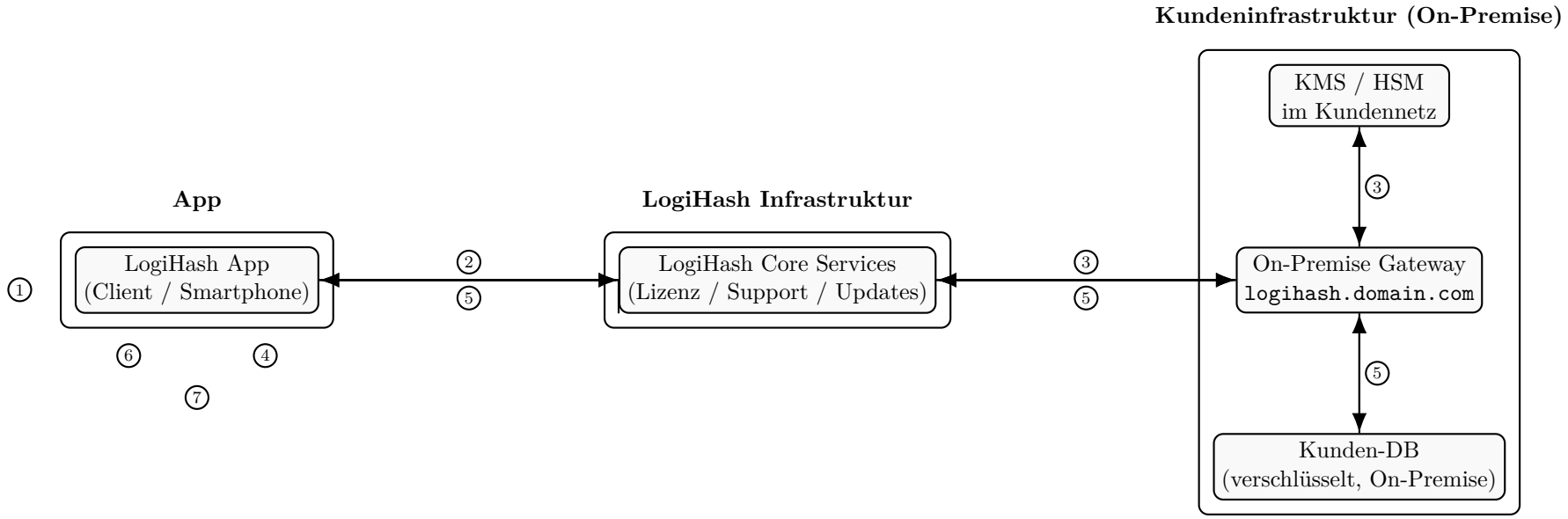
3. Cloud-basierte LogiHash-Lösung – Verifikationsfluss beim T-Code-Scan



Prozessbeschreibung – Cloud-basierte Lösung (Schritte 1–7):

1. **Scan & Parsing:** Die LogiHash App scannt den T-Code und parst die Hauptkomponenten: Creator ID, DB Entry ID, Unique Symmetric Key, KID und digitale Signaturen.
2. **Schlüsselanforderung:** Die App sendet eine Anfrage an `api.logihash.com`, um anhand von Creator ID und KID den passenden Public Key zu erhalten.
3. **Schlüsselbereitstellung:** Das API-Gateway fragt KMS/HSM ab und erhält den passenden Public Key, der über das Gateway an die App zurückgesendet wird.
4. **Signaturprüfung (lokal):** Die App verwendet den Public Key, um die digitale Signatur des T-Codes lokal zu verifizieren.
5. **Datenanforderung:** Nach erfolgreicher Signaturprüfung sendet die App eine Anfrage mit Creator ID und DB Entry ID an `api.logihash.com`, um die verschlüsselten Daten hinter dem T-Code abzurufen.
6. **Lokale Entschlüsselung:** Die App entschlüsselt die erhaltenen Daten lokal mit dem Unique Symmetric Key, der ausschließlich im T-Code gespeichert ist (wird serverseitig nicht vorgehalten).
7. **Anzeige:** Die App zeigt die entschlüsselten Informationen dem Nutzer an.

4. On-Premise LogiHash-Lösung – Verifikationsfluss beim T-Code-Scan



Prozessbeschreibung – On-Premise Lösung (Schritte 1–7):

1. **Scan & Parsing:** Die LogiHash App scannt den T-Code und parst die Hauptkomponenten: Creator ID, DB Entry ID, Unique Symmetric Key, KID und digitale Signaturen.
2. **Schlüsselanforderung:** Die App sendet die Anfrage zunächst an `api.logihash.com`.
   Die LogiHash Core Services empfangen die Anfrage und leiten sie an die kundenspezifische On-Premise-Umgebung (`logihash.domain.com`) weiter, um dort anhand von Creator ID und KID den passenden Public Key abzufragen.
3. **Schlüsselbereitstellung:** Das On-Premise Gateway fragt das KMS/HSM im Kundennetz ab, erhält den passenden Public Key und liefert ihn über die LogiHash Core Services an die App zurück.
4. **Signaturprüfung (lokal):** Die App verwendet den Public Key, um die digitale Signatur des T-Codes lokal zu verifizieren.
5. **Datenanforderung:** Nach erfolgreicher Signaturprüfung sendet die App eine Anfrage mit Creator ID und DB Entry ID an `api.logihash.com`.
   Die LogiHash Core Services leiten diese Anfrage an die kundenspezifische On-Premise-Umgebung (`logihash.domain.com`) weiter, wo das On-Premise Gateway auf die verschlüsselte Kunden-DB zugreift, den Datensatz abruft und über Core und Gateway zurück an die App liefert.
6. **Lokale Entschlüsselung:** Die App entschlüsselt die erhaltenen Daten lokal mit dem Unique Symmetric Key, der ausschließlich im T-Code gespeichert ist.
7. **Anzeige:** Die App zeigt die entschlüsselten Informationen dem Nutzer an.