**1. Section: “The plangroups startup logic will be skipped…”**

**Location:** Near the end of the “Solution Sketches” section (after the application startup sequence description, before “Service Logic”).  
**Current Text to Delete:**

The plangroups startup logic will be skipped if prior requests cached the plangroups for this id\_token in CVS plangroups cache.

**Replace With (from CVS BFF Shared Documentation):**

The plangroups startup logic will be skipped if the CVS BFF detects that a valid Redis cache entry exists for the current user session.  
The cache key follows the standardized CVS BFF key generation pattern:

* **Prefix:** cvsbff:plangroups
* **Proxy Resource ID:** Extracted from eieheaderusercontext.eieHeaderBusinessIdentifier where idSource === 15
* **Impersonation Account ID:** If applicable, appended from accountIdentifier.idValue in EIE headers  
  Example: cvsbff:plangroups:MXN76FFFFPXZ:QA2-AYANAX

This ensures data isolation across users and impersonated sessions, and prevents redundant backend calls. Redis TTL values should align with the CVS BFF configuration (CACHE\_TTL), and cache invalidation must occur on membership changes.

**2. Section: “3.a Cache Check” in Service Logic**

**Location:** In the “Service Logic” section, step 3.a under “BFF Orchestration” for /sa/plandocs/v1/list.  
**Current Text to Delete:**

3.a Cache Check  
3.a.1 Checks cache using the same cache key as ID Cards (EIE Header EIEHeaderUserContext)  
3.a.2 If cached, skip to step 5

**Replace With (from CVS BFF Shared Documentation):**

3.a Cache Check  
3.a.1 The CVS BFF generates a cache key using the CACHE\_PREFIX, the relevant domain (planGroups), and the account key composed of accountId and impersonatingAccountId (if present).  
3.a.2 Example format: {CACHE\_PREFIX}:{domain}:{accountKey} → e.g., cvs-bff:prod:planGroups:12345admin123.  
3.a.3 The BFF queries Redis using this key. If data is found, it bypasses backend calls and returns the cached payload.  
3.a.4 Cache entries are TTL-bound (CACHE\_TTL env variable) and separated per environment via CACHE\_PREFIX.

**3. Section: “6.d Caching” in Service Logic**

**Location:** Still in the “Service Logic” section, step 6.d under “Features Check”.  
**Current Text to Delete:**

6.d Caching  
If features are enabled, CVS-BFF caches the trimmed down list using the same cache key as ID Cards

**Replace With (from CVS BFF Shared Documentation):**

6.d Caching  
If features are enabled, the CVS BFF stores the filtered plan groups or document list in Redis.

* **Cache Key:** Built using the standard CVS BFF buildCacheKey() method, incorporating the domain (planGroups or other relevant domain), the accountId, and the impersonatingAccountId when applicable.
* **Example:** cvs-bff:dev:planGroups:12345 or cvs-bff:dev:planGroups:12345admin123
* **TTL:** Determined by CACHE\_TTL (in seconds) and configurable per environment.  
  This approach maintains isolation per account and per impersonation session, ensuring correct data retrieval in multi-user contexts.