Discussion Questions

1. **Eventual Consistency Concept in this context**

In the `SecurityDashboardModule`, now, **eventual consistency** just means that if someone changes their password in the `UserManagementModule`, you might not see that change right away on the dashboard. There's always some **delay** because the system needs to wait for the **Kafka event** to go through. So, there's a small gap between when the password actually gets changed and when the **new timestamp** shows up on the dashboard. Having this **eventual way** of handling things is a better approach to keep the system running smoothly, instead of making everything wait to update at the same time.

2. **What potential challenges or considerations arise from this for the data displayed in the SecurityDashboardModule?**

The `SecurityDashboardModule` has a few basic problems that come from how event-driven systems usually work. For example, because it depends on events, sometimes the password change times it shows are already a few seconds or even minutes behind. This can **confuse admins** who expect the info to update almost instantly. Even worse, if something goes wrong — like the Kafka brokers go down or there's a **network issue** — the events might not even reach the dashboard at all. In that case, it just keeps showing the **wrong timestamp** until someone goes in and fixes it manually. There's also the issue of **message order**. If a user changes their password quickly a few times, those events can show up in the wrong order, so the dashboard might display an older password time like it's the latest one. That makes it hard to tell what really happened last. On top of that, **duplicate events** can show up too — if Kafka restarts or retries messages, the same event might be sent more than once. And if we don't have something in place to **catch and skip duplicates**, we can end up **messing up the timestamp history** even more.

3. **What if a strictly consistent view across all modules were required for a specific operation (no implementation, just considerations)?**

Sometimes, we'd prefer that all modules show **exactly the same data at the same time**, like during a **security audit** or **compliance check**. There are a few ways to handle this. One is the **direct query approach**, where if we really need accurate info, the `SecurityDashboardModule` can directly ask the `UserManagementModule` instead of relying on its own stored or cached data. Another method is using **synchronized updates**, where both modules are updated at the same moment — but this can **slow things down** and make the system less available. We could also add a **manual refresh option** that pulls the latest data from the main module whenever someone needs it. The smartest thing to do in most cases is to stick with the **fast eventual consistency model** for regular usage, and only switch to **direct verification** when we're generating important security reports or audit logs, where showing the **exact time** really matters for business needs.