Tokeneer case study - SHARCS development

1 Overview

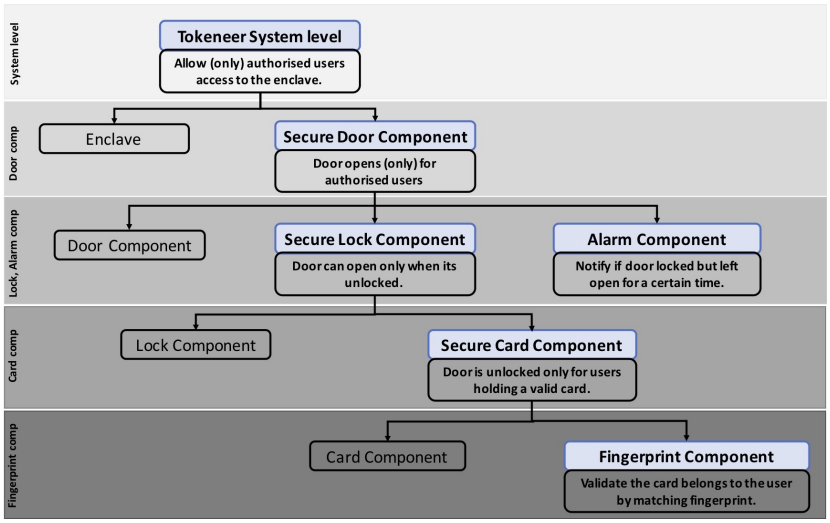

Figure 1: Tokeneer: hierarchical component design, flow down requirements

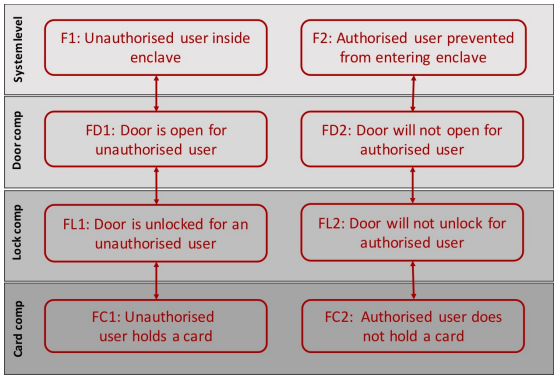

Figure 2: Tokeneer: hierarchical failures

2 System Level

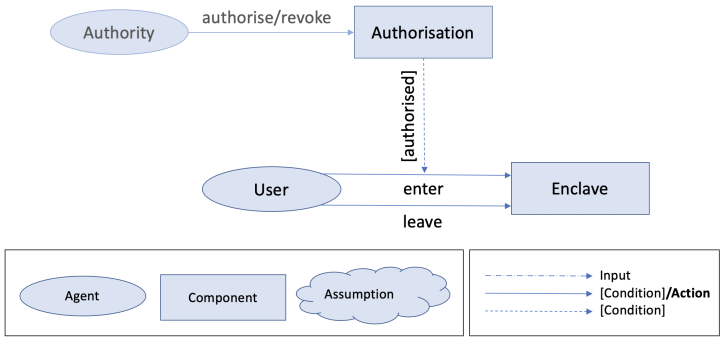

Figure 3: System level, control abstraction diagram

System level			
Purpose: Allow (only) authorised users access to the enclave. **Actions:** Users can enter and leave enclave.			
Failures: • **F1**: Unauthorised user inside enclave • **F2**: Authorised user prevented from entering enclave			
System Action	**Not Occurring Causes Failure**	**Occurring Causes Failure**	**Wrong Timing or Order Causes Failure**
User Enter Enclave	**A11:** Authorised user prevented from entering enclave (*F2*)	**A12:** Unauthorised user enters enclave (*F1*)	N/A
User Leave Enclave	No failure	No failure	N/A
Mitigations: • ***Door*** component opens (only) for authorised users (addressing *A11, A12*)			

Figure 4: System level, action analysis table

3 Door Component

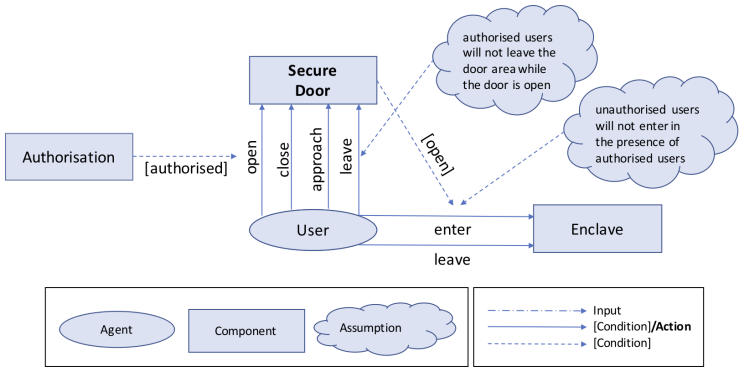

Figure 5: Door component, control abstraction diagram

Door Component

Purpose: Door opens (only) for authorised users.
Actions: Users can open and close doors.

Failures:

- **FD1**: Door is open for unauthorised user (causes *F1*)
- **FD2**: Door will not open for authorised user (causes *F2*)

System Action	Not Occurring Causes Failure	Occurring Causes Failure	Wrong Timing or Order Causes Failure
User Open Door	**AD11:** Authorised user is unable to open the door (*FD2*).	**AD12:** Unauthorised user opens the door (*FD1*)	N/A
User Close Door	**AD21:** User does not close the door (*FD1*)	No failure	**AD23:** Authorised user closes door before entering (*FD2*)
User Approach Door	No failure	No failure	No failure
User Leave Door	No failure	No failure	**AD43**: Authorised user leaves door, when door is open, and so the door is left open for an unauthorised user

Mitigations:

- ***Lock*** component controls when the door can be opened (addressing *AD11, AD12*)
- ***Alarm*** component warns when door is left open for a certain time (addressing *AD21*)
- If a user closes the door before entering, they can open it again (addressing *AD23*)
- Authorised users will not leave the door area while the door is open (addressing AD43)

Figure 6: Door component, action analysis table

4 Look Component

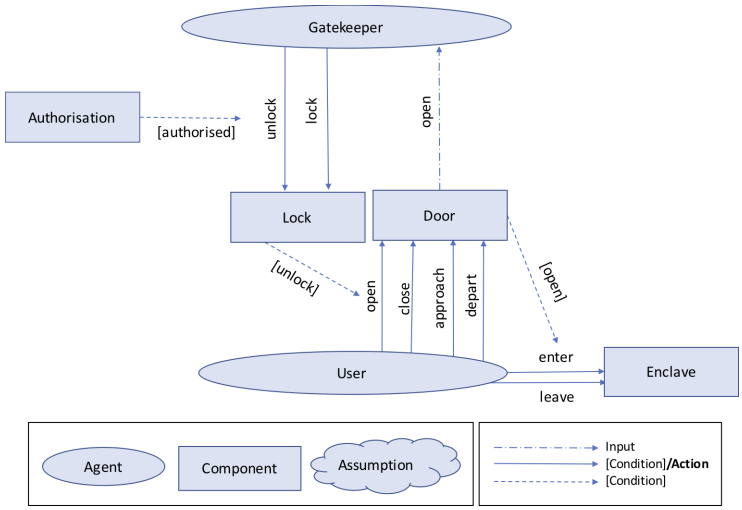

Figure 7: Lock component, control abstraction diagram

Lock Component

Purpose: Door can open only when its unlocked.
Actions: Door can lock and unlock for users.

Failures:

- **FL1**: Door is unlocked for an unauthorised user (causes *FD1* and so *F1*)
- **FL2**: Door remains locked for an authorised user (causes *FD2* and so *F2*)

System Action	Not Occurring Causes Failure	Occurring Causes Failure	Wrong Timing or Order Causes Failure
Unlock Door	**AL11:** Door remains locked for an authorised user (*FL2*)	**AL12:** Door unlocks for an unauthorised user (*FL1*)	N/A
Lock Door	**AL21:** Door remains unlocked for an unauthorised user (*FL1*)	N/A	**AL23:** Door locks before user opens door (*FL2*)

Mitigations:

- ***Card*** component door is unlocked only for users holding a valid card (addressing *AL11*, *AL12*)
- Lock component is verified to automatically re-lock when the door closes (addressing *AL21*)
- Lock component is validated to give sufficient time before automatically re-locking (addressing *AL23*)

Figure 8: Lock component, action analysis table

5 Alarm Component

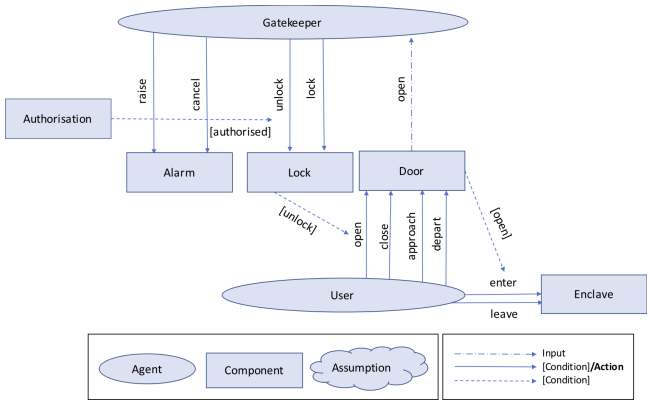

Figure 9: Alarm component, control abstraction diagram

Alarm Component

Purpose: Notify if door locked but left open for a certain time.
Actions: Alarm can start or clear.

Failures:

- **FA1:** Alarm off when door is left open for a certain time (leading to *FD2* and so *F1*)
- **FA2:** Alarm on when door is closed or soon after door opened (this may lead to alarm notifications being ignored, hence leading to *FD2* and so *F1*)

System Action	Not Occurring Causes Failure	Occurring Causes Failure	Wrong Timing or Order Causes Failure
Alarm Start	**AA11:** Alarm does not start when door is left open (*FA1*).	**AA12:** Alarm starts when door is closed (*FA2*)	**AA13a:** Alarm started too late means that door is left open without notification for too long (*FA1*). **AA13b:** Alarm started too quickly after door opened (*FA2*)
Alarm Clear	**AA21:** Alarm does not stop after door is closed (*FA2*)	N/A	**AA23a:** Alarm cleared too quickly means that door is left open without notification (*FA1*). **AA23b:** Alarm cleared too late may (*FA2*)

Mitigations:

- Alarm component is verified to ensure that it starts when the door is left open for a certain time and stops as soon as the door is closed is always given correctly (addressing *AA11, AA12, AA21, AA23a, AA23b)*
- The time delay between opening the door and starting the alarm must be chosen by validation and experimentation involving domain experts (addressing *AA23a, AA23b*)

Figure 10: Alarm component, action analysis table

6 Card Component

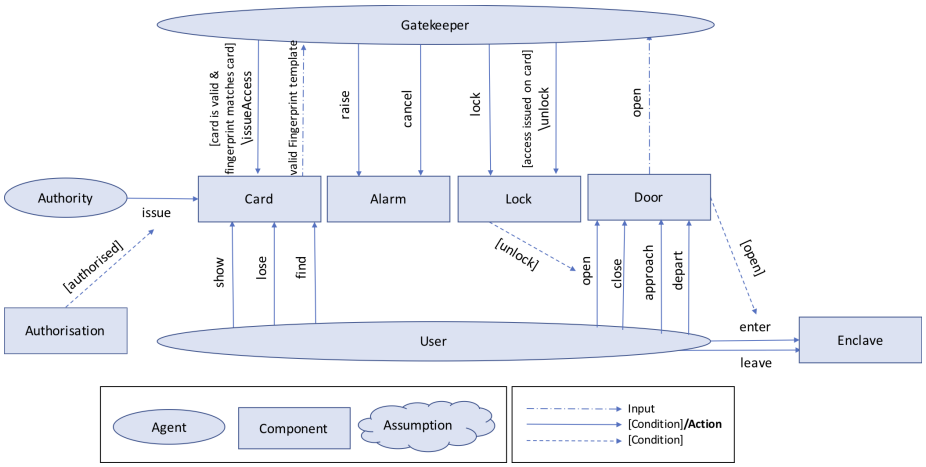

Figure 11: Card component, control abstraction diagram

Card Component			
Purpose: Door is unlocked only for users holding a valid card. **Actions:** Card can be issued for a user.			
Failures: • **FC1**: Unauthorised user holds a card (causes *FL1* and so *FD1* and *F1*) • **FC2**: Authorised user does not hold a card (causes *FL2* and so *FD3* and *F2*)			
System Action	**Not Occurring Causes Failure**	**Occurring Causes Failure**	**Wrong Timing or Order Causes Failure**
Issue Card	**AC11:** Authorised user not issued a card (*FC2*)	**AC12:** Unauthorised user is issued a card (*FC1*)	N/A
Lose Card	No failure	**AC22:** Authorised user loses card (*FC2*)	N/A
Find Card	No failure	**AC32:** Unauthorised user finds card (*FC1*)	N/A
Mitigations: • Out of scope – an authorisation authority will deal with users without cards (addressing *AC11, AC22)* • ***Fingerprint*** component ensures door is unlocked only for users with a fingerprint that matches the card that they hold (addressing *AC12, AC32*)			

Figure 12: Card component, action analysis table

7 Fingerprint Component

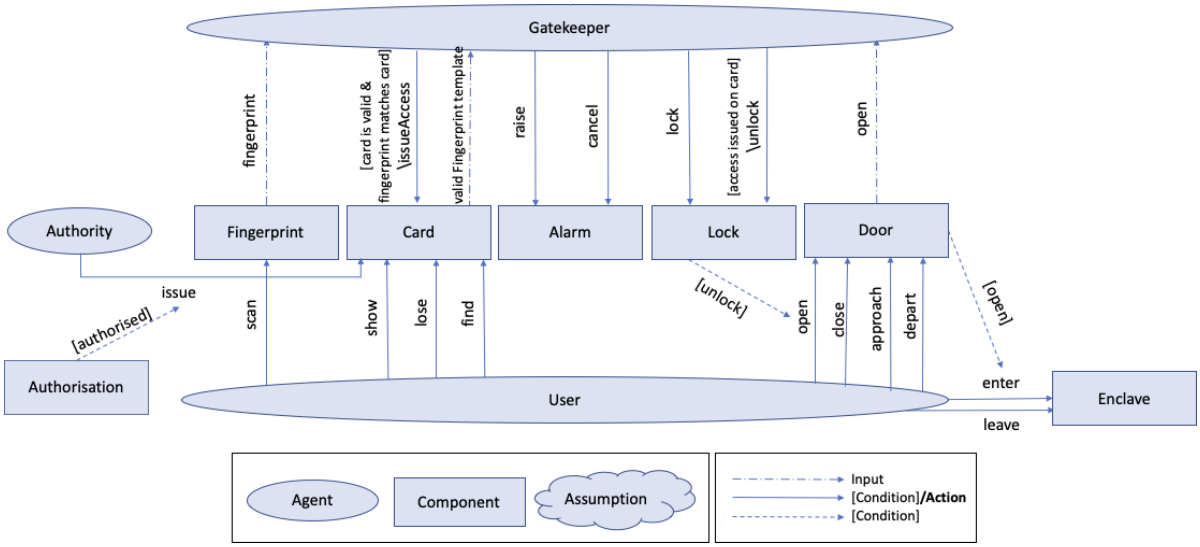

Figure 13: Fingerprint component, control abstraction diagram

Fingerprint Component

Purpose: Validate the card belongs to the user by matching fingerprint.
Actions: The fingerprint on the card is compared with the user's fingerprint and if a match is found, the card is valid.

Failures:

- **FF1:** Authorised user does not hold validated card (new failure leading to *F1*)
- **FF2:** Unauthorised user has validated card (causes *FC1* and so *FL1, FD1, F1*)

System Action	Not Occurring Causes Failure	Occurring Causes Failure	Wrong Timing or Order Causes Failure
Match Fingerprint	**AF11:** Authorised users card is not validated (*FF1*)	**AF12:** Card is incorrectly validated for an unauthorised user (*FF2*)	**AF13:** Card is validated after the lock is unlocked

Mitigations:

- Fingerprint component is verified to ensure that validation is always given correctly (addressing *AF11, AF12)*
- Lock component is verified to ensure that it cannot unlock without the card being validated (addressing *AF13)*

Figure 14: Fingerprint Component, action analysis table