



X-SCAN

**INTEGRATION IN
THIRD PARTY APPS**



SCOPE

This document presents the 3 possible options to integrate the X-SCAN support within 3rd party applications :

- Option 1 - Code content injection
- Option 2 - Crosscall intent
- Option 3 - SDK integration

The benefits and limitations of each method will be precised hereafter



X-SCAN USE STEPS

The use of X-SCAB with a 3rd party application requires the following steps

**X-SCAN
CONFIGURATION**

**SCAN OPERATION
TRIGGER**

**SCANNED CONTENT
HANDLING**

These steps will be managed differently depending the integration mode chosen



OPTION 1 : CODE CONTENT INJECTION

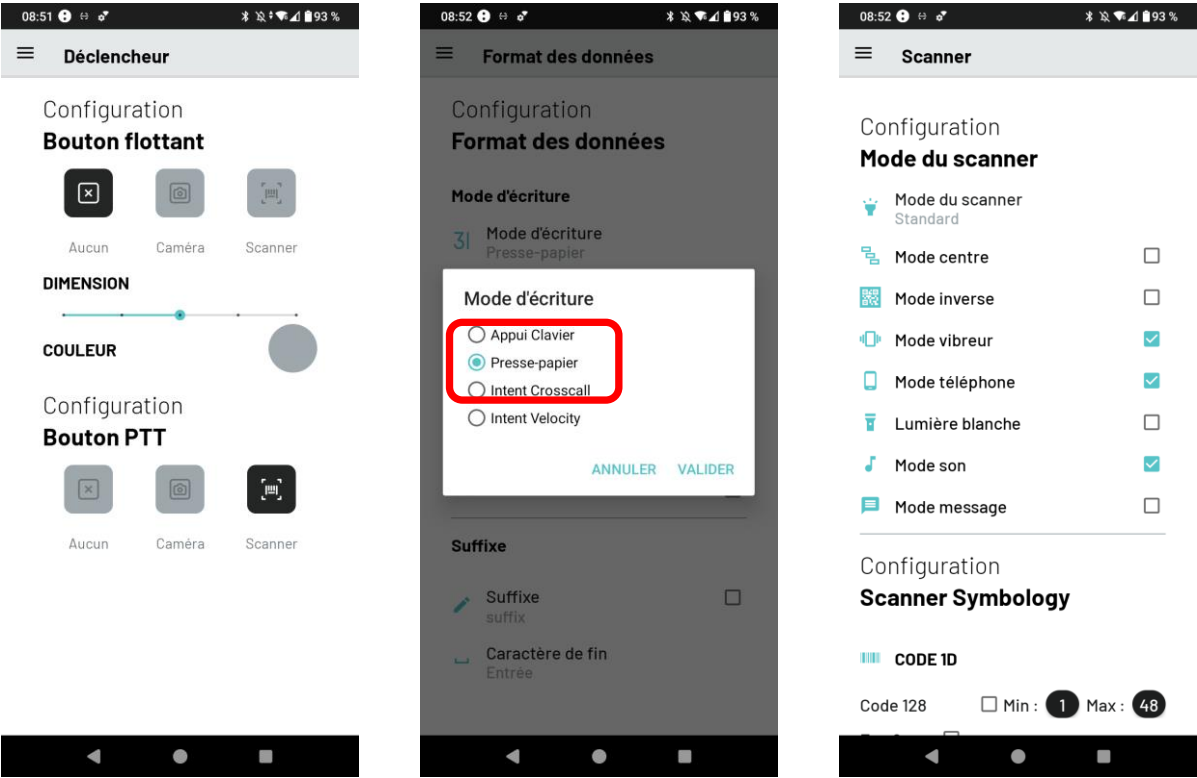
Principle

- No integration required in the 3rd party application
- X-TRACK application required for the X-SCAN configuration and control
- The scan can be triggered either with the programmable button or with a floating button
- The content of the scanned code is inserted in the text field which has focus on the smartphone
- 2 injections mode supported :
 - Keyboard press (More reliable but slower)
 - Copy/Paste (Faster but doesn't work with some apps)



OPTION 1 : CODE CONTENT INJECTION

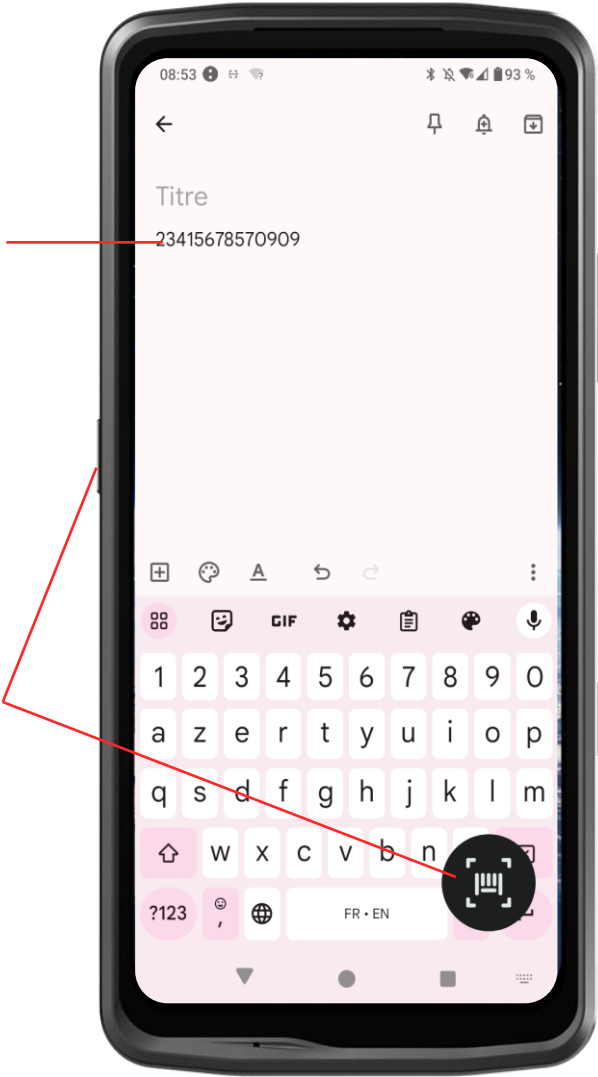
User Experience



X-SCAN configuration through X-TRACK application

Scanned content
inserted in the
selected text field

Press programmable
button or floating
button





OPTION 2 : CROSSCALL INTENT

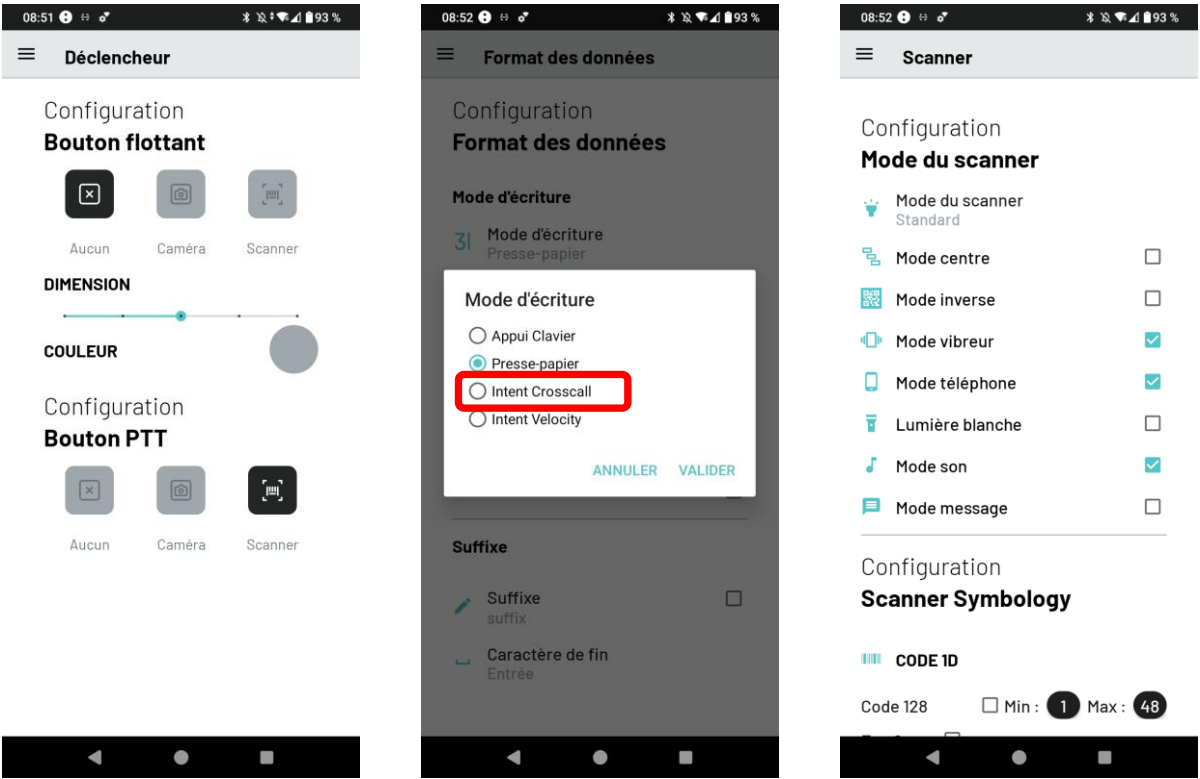
Principle

- Limited integration required in the 3rd Party application
- X-TRACK application required for the X-SCAN configuration and control
- The scan can be triggered either with the programmable button or with a floating button
- An intent message is broadcasted each time a code is scanned
 - ACTION : `com.crosscall.xtrack.action.BARCODE`
 - VARIABLE : `com.crosscall.xtrack.data_string`
- The 3rd party application catches this message and implements the related logic



OPTION 2 : CROSSCALL INTENT

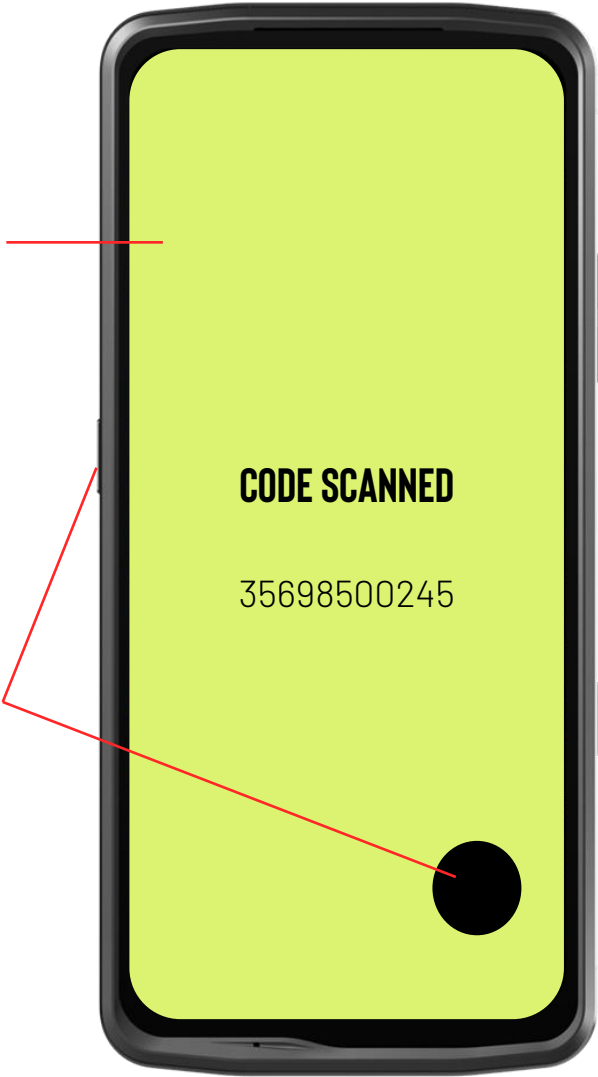
User Experience



X-SCAN configuration through X-TRACK application

3rd party app catches
the intenb message
and implement the
related logic

Press programmable
button or floating
button





OPTION 3 : SDK INTEGRATION

Principle

The X-SCAN is fully managed by the 3rd party app for:

- The X-SCAN configuration
- The scan event trigger
- The scanned content handling

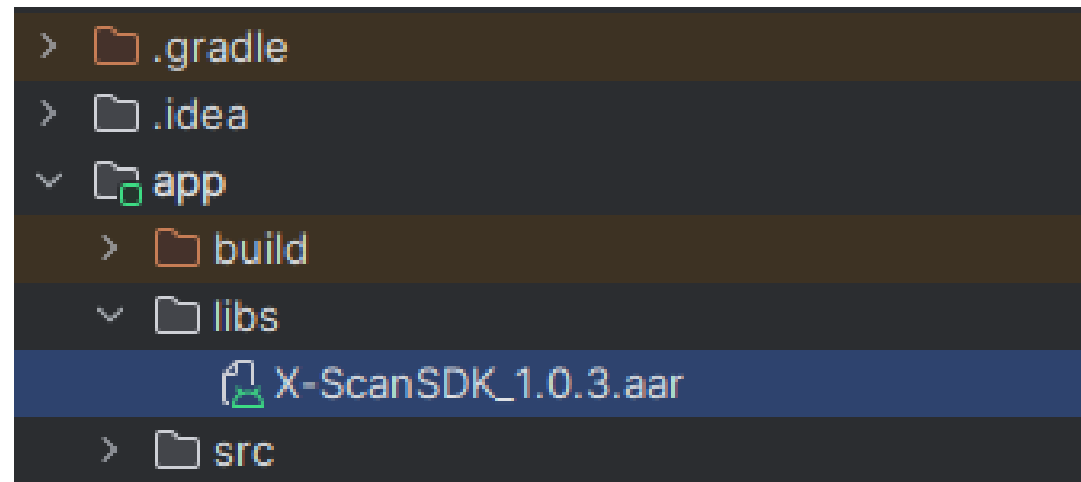
X-TRACK application is not required



OPTION 3 : SDK INTEGRATION

Library integration

*.AAR addition to Android STUDIO





OPTION 3 : SDK INTEGRATION

Initialization = SDK and X-SCAN configuration

- Symbolologies (Types & Length)
- Read mode (Level, Pulse, Sense, Batch)
- Center mode
- Reticule
- Audio and vibration settings
- Timer
- Sensitivity

```
private val xScanManager = XScanManager(appContext, listener: this)

init {
    xScanManager.initialize(
        disconnectedBeep = false,
        scanBeep = false,
        connectBeep = false,
        showToast = false,
        vibration = true
    )
}
```

```
override fun startConnexion() {
    startUsbConnection()
}
```



OPTION 3 : SDK INTEGRATION

Initialization = SDK and X-SCAN configuration

```
private fun startUsbConnection() {  
    val formats: MutableList<IBarcodeFormat> = mutableListOf()  
    xScanManager.connect()  
    formats.add(BarcodeFormatAztec(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatCode39(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatCode93(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatCode128(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatDataMatrix(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatEan8(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatEan13(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatFebraban(IBarcodeFormat.ActivationFebraban.Enable))  
    formats.add(BarcodeFormatGS1_128(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatGS1Databar(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatIndustrial25(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatISBN(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatISSN(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatITF20f5(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatITF6(IBarcodeFormat.ActivationITF.EnableAndTransmitCheckCharacter))  
    formats.add(BarcodeFormatITF14(IBarcodeFormat.ActivationITF.EnableAndTransmitCheckCharacter))  
    formats.add(BarcodeFormatMatrix20f5(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatMicroQrCode(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatMSIPlessey(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatPdf417(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatPlessey(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatQr(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatStandard25(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatUPCA(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatUPCE(IBarcodeFormat.Activation.Enable))  
    xScanManager.setBarcodeFormatEnable(formats)  
}
```



OPTION 3 : SDK INTEGRATION

Activation (depends from the selected mode)

- Level Mode : Instantaneous capture / Active during 5 sec
- Pulse mode : Instantaneous capture / Active as long as the button is pressed
- Sense mode : Movement detection
- Batch mode : Lot detection / Replicat exclusion

```
override fun startScan() {  
    xScanManager.startScan()  
}
```

Unique command whatever the selected mode



OPTION 3 : SDK INTEGRATION

Callbacks handling

Handling of scan reading event proper to each 3rd Party application

```
override fun onScanResult(result: String) {  
    scope.launch { this: CoroutineScope  
        _usbEventFlow.emit(result)  
    }  
}
```



OPTION 3 : SDK INTEGRATION

Desactivation

X-SCAN deconnexion

```
override fun stopConnexion() {  
    xScanManager.close()  
}
```



SYNTHESIS

X-TRACK	OPTION 1 (INJECTION)	OPTION 2 (INTENT)	OPTION 3 (SDK)
X-SCAN configuration	Managed by X-TRACK	Managed by X-TRACK	Managed by 3rd party app
Scan trigger	Managed by X-TRACK	Managed by X-TRACK	Handled by 3rd party app
Scanned code handling	Managed by X-TRACK	Handled by 3rd party app	Handled by 3rd party app
Integration effort	None	Limited (1 to 2 days)	Simple (2 to 3 days)
Advantages	<ul style="list-style-type: none">No modification required of 3rd party appCan be used with any existing native or web app through a simple text field	<ul style="list-style-type: none">Limited impact on 3rd party appDo not requires an app with text field	<ul style="list-style-type: none">No need to handle X-TRACK deployment and configurationBetter scan performancesFull and dynamic control of X-SCAN configuration (eg to activate torch, screen mode...)Full control on UX
Target	<ul style="list-style-type: none">Small/Medium companies without programming teamsTest purposes	<ul style="list-style-type: none">Companies with programming teams	<ul style="list-style-type: none">Companies with programming teams