

안드로이드 애플리케이션의 프로세스

다이스

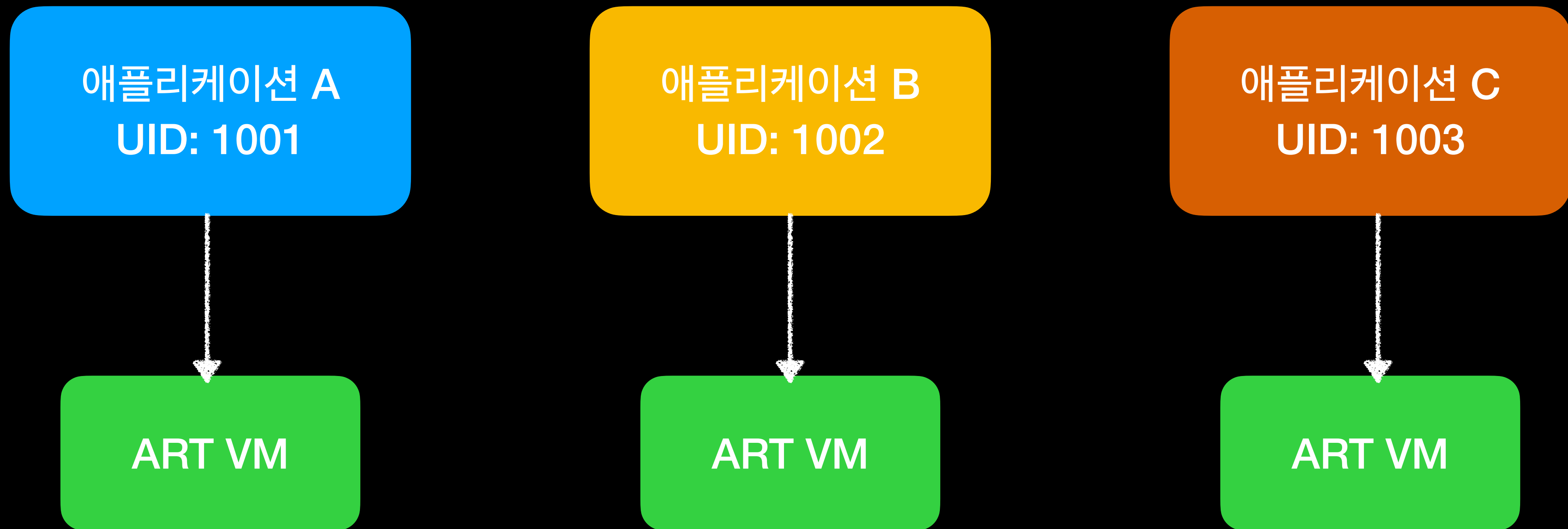
안드로이드 애플리케이션 개요

앱은 어떻게 실행될까? - 프로세스 구조

- 안드로이드 애플리케이션은 리눅스 커널 기반에서 실행
- 애플리케이션마다 `fork()` 호출로 독립 프로세스를 생성
- 각 애플리케이션은 고유 UID를 가지고 있음
- 별도 메모리 공간에서 샌드박스 방식으로 실행
- Android Runtime(ART) 또는 Dalvik 가상 머신 위에서 격리된 환경 제공

안드로이드 애플리케이션 개요

앱은 어떻게 실행될까? - 프로세스 구조



안드로이드 애플리케이션 개요

앱은 어떻게 실행될까? - 프로세스 구조

애플리케이션 간 데이터 보호

시스템 안정성 향상

충돌 격리 및 보안 강화

컴포넌트 프로세스 분리

컴포넌트별로 프로세스를 나눌 수 있다

- 기본적으로 모든 컴포넌트는 애플리케이션 프로세스 내에서 실행됨
- 하지만 android:process 속성을 사용해 별도 프로세스에서 실행 가능
- 같은 접두사 (:remote 등)를 공유하면 같은 별도 프로세스 사용

```
<service  
    android:name=".MyService"  
    android:process=":remote" />
```

컴포넌트 프로세스 분리

컴포넌트별로 프로세스를 나눌 수 있다

- :remote는 앱 내부의 새로운 프로세스
- 애플리케이션 간 공유를 위해서 동일 UID + 서명이 필요

```
<service  
    android:name=".MyService"  
    android:process=":remote" />
```

```
[Main Process]  
└─ MainActivity  
   OtherActivity
```

```
[:remote Process]  
└─ MyService
```

프로세스 우선순위 구조

시스템은 어떤 프로세스를 먼저 죽일까?

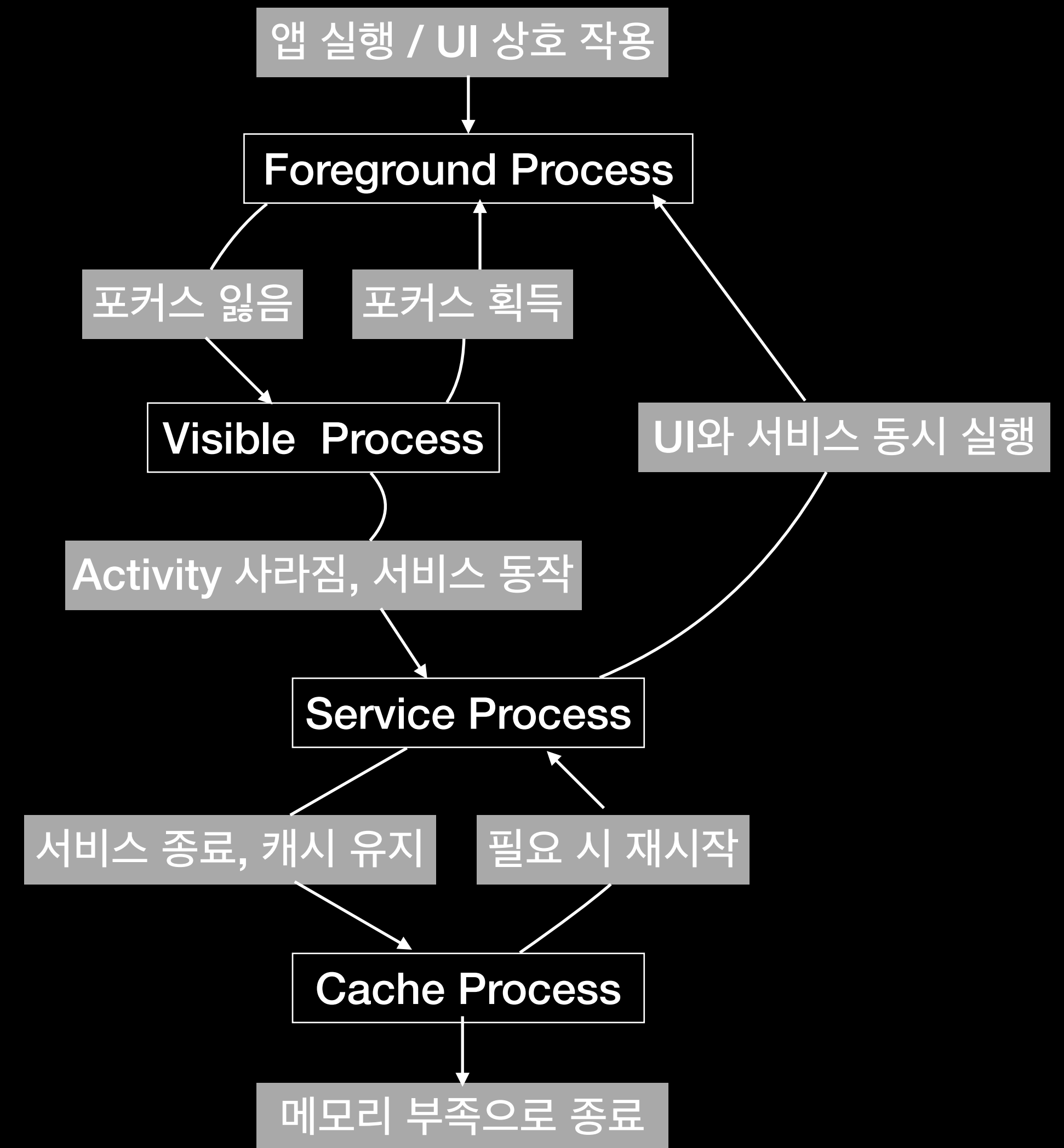
안드로이드는 메모리 부족 시 아래 순서대로 프로세스를 종료합니다

우선순위	유형	설명
1	Foreground Process	사용자와 상호작용 중인 Activity 등
2	Visible Process	보이지만 포커스 없는 UI (예: Dialog 뒤)
3	Service Process	UI 없는 백그라운드 작업
4	Cached Process	백그라운드 종료 상태로 메모리만 유지

프로세스 상태 전이

상태는 어떻게 전이될까?

사용자 동작 또는 시스템 자원 상태에 따라
프로세스 상태가 전환됨



보안 모델과 샌드박싱

UID와 권한 기반의 보안 모델

- 애플리케이션은 고유 UID 기반으로 실행
- 다른 애플리케이션의 데이터/파일/프로세스에 접근하려면 권한 필요
- Android의 권한 시스템 + 리눅스 커널의 격리 -> 샌드박싱 환경 구성
- 애플리케이션 A와 애플리케이션 B는 직접 통신/접근 불가
- ContentProvider, IPC, FileProvider 사용 시 별도 허용 필요함

실전 전략 - 프로세스 분리

언제 컴포넌트를 별도로 실행할까?

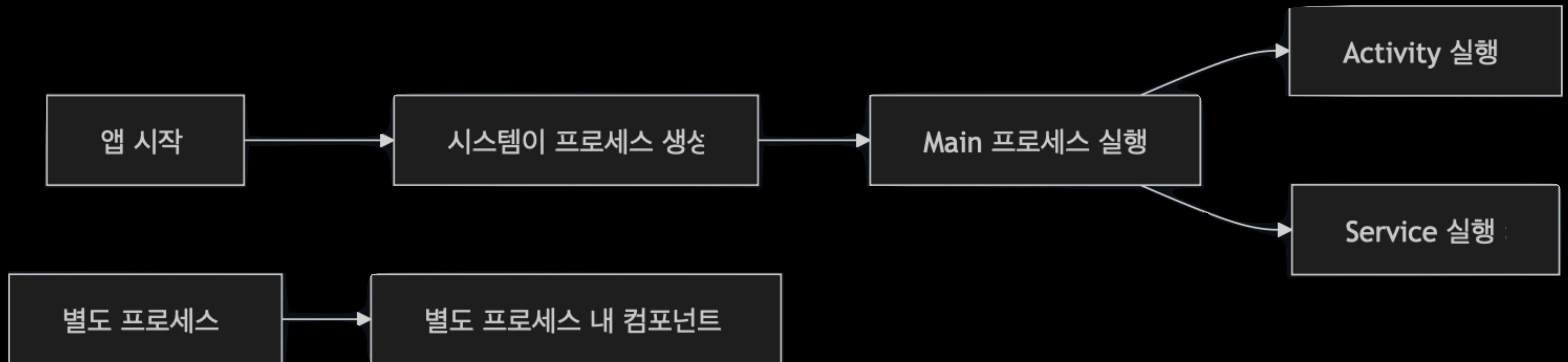
- 무거운 컴포넌트 분리 (WebView, 이미지 렌더링 등)
- 서비스의 독립 생존 필요 (백그라운드 지속성 보장)
- 메인 애플리케이션 충돌 시에도 서비스 유지 가능

```
<service  
    android:name=".HeavyService"  
    android:process=":heavy" />
```

실행 흐름 시나리오

프로세스 실행 흐름 요약

- 애플리케이션 런처 클릭 -> 시스템이 필요 시 프로세스/컴포넌트 생성
- 각 컴포넌트는 메인 또는 별도 프로세스에서 실행됨



핵심 요약

지금까지의 핵심 요점 정리

- 안드로이드는 각 애플리케이션을 독립 프로세스 + 샌드박스 환경에서 실행
- 컴포넌트는 android:process로 분리 실행 가능
- 프로세스는 우선순위 기반으로 생명주기 관리
- 개발자는 구조적으로 프로세스 분리와 복구를 설계해야 함