Яндекс

Рождение, жизнь и смерть,

или что происходит с приложением в системе

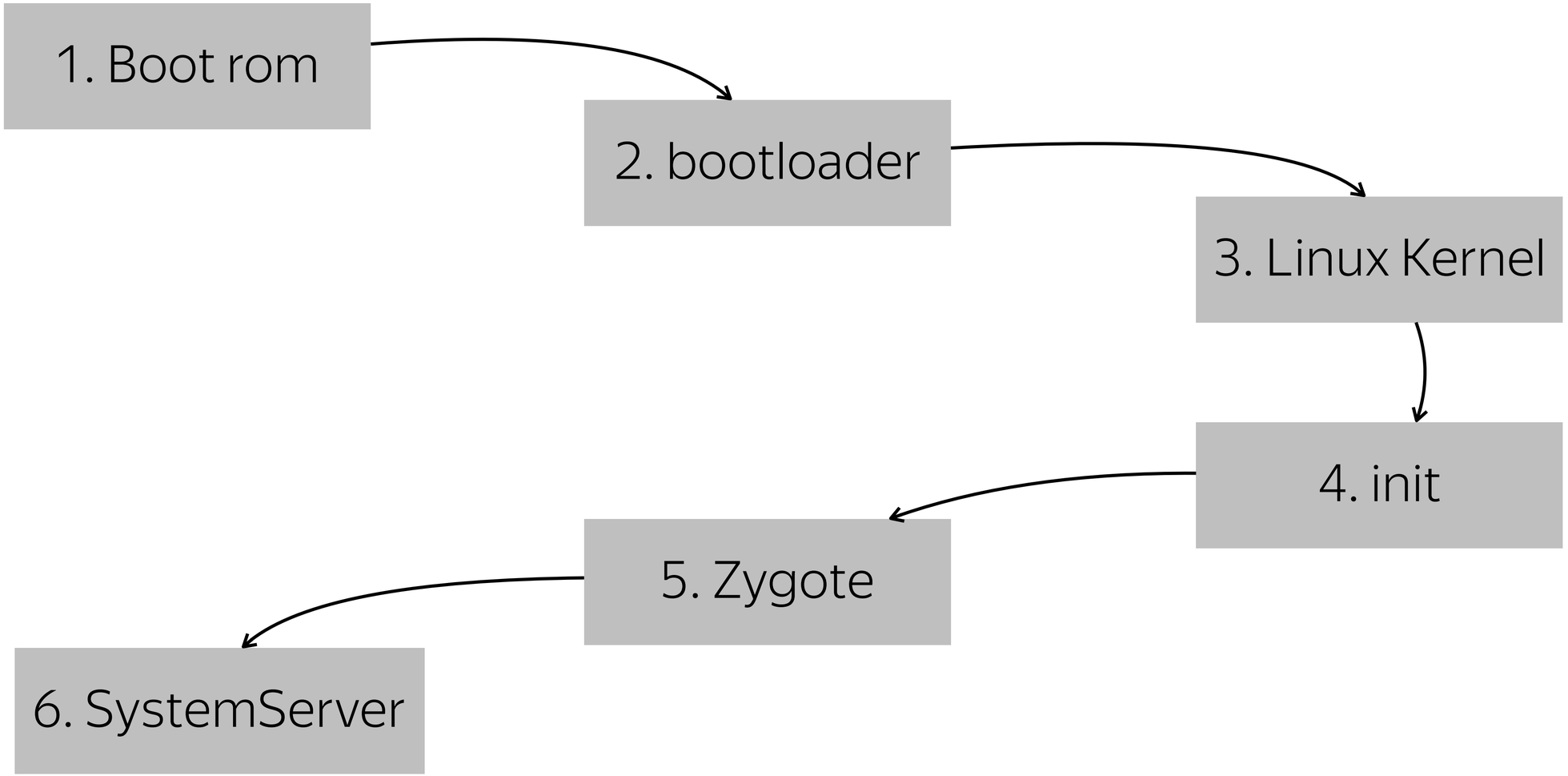
Антон Дудаков
Лаборатория встраиваемых автомобильных решений

О чём пойдёт речь

- Загрузка Android
- Старт приложения
- Взаимодействие приложения с системой
- Завершение и возобновление работы приложения

Загрузка Android

Старт системы



boot rom

- Аналог BIOS
- Нужен чтобы запустить bootloader

boot loader

Первая программа

подключает образ с Linux kernel,
инициирует его запуск

обычно умеет управлять разделами,
а также выполнять обновление

Linux Kernel

- Инициализируется система ввода/вывода
- Загружаются драйвера
- Монтируется корневая файловая система
- Запускается первый процесс ОС – init

init

- Прочитываются init.rc файлы, в которых описаны конфиги того, что нужно делать при загрузке
- Монтируются файловые системы
- Стартуют несколько системных демонов (управление дисками, сетью, временем)
- Запускается Zygote

Zygote

service zygote /system/bin/app_process -Xzygote /system/bin --zygote —start-system-server

Zygote.forkSystemServer

Зигота является тотипотентной клеткой, то есть способной породить любую другую

https://ru.wikipedia.org/wiki/Зигота

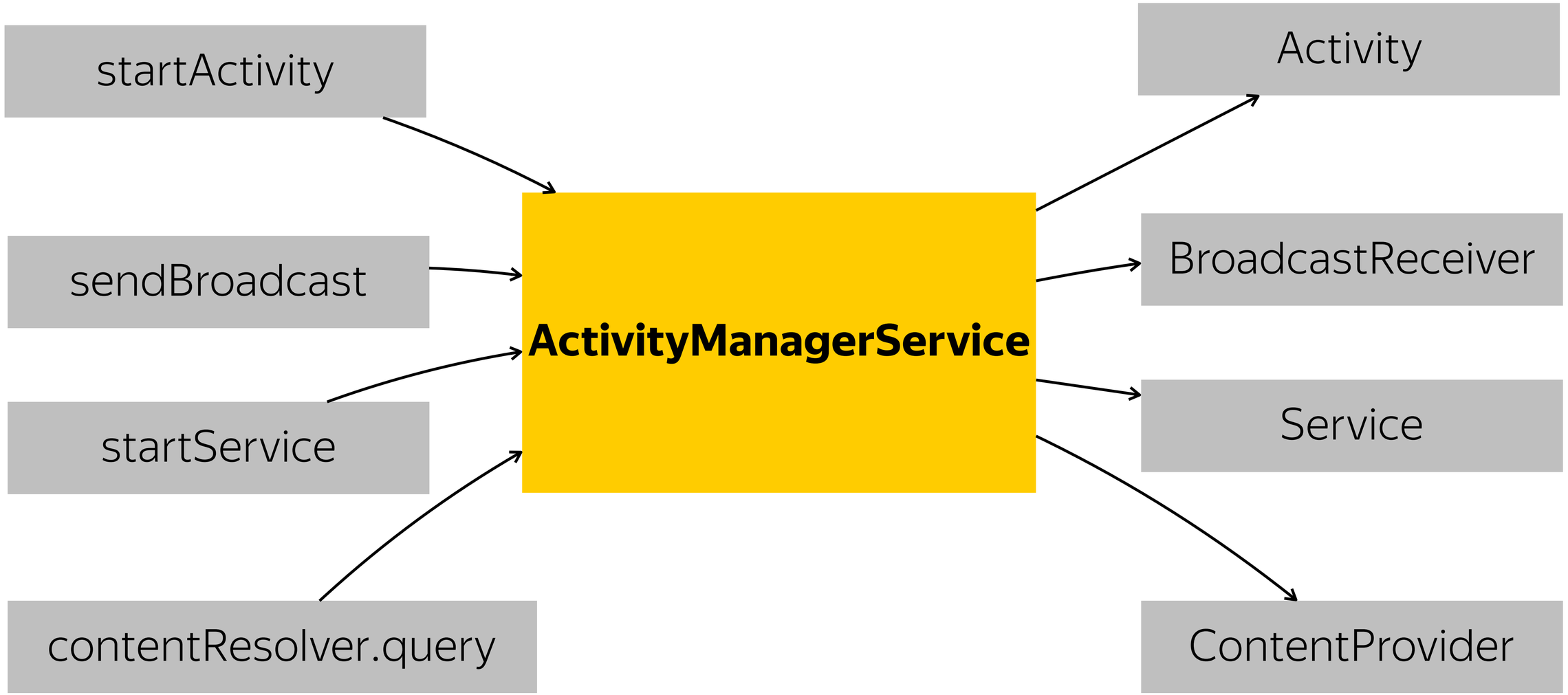
Первое Android-приложение

SystemServer

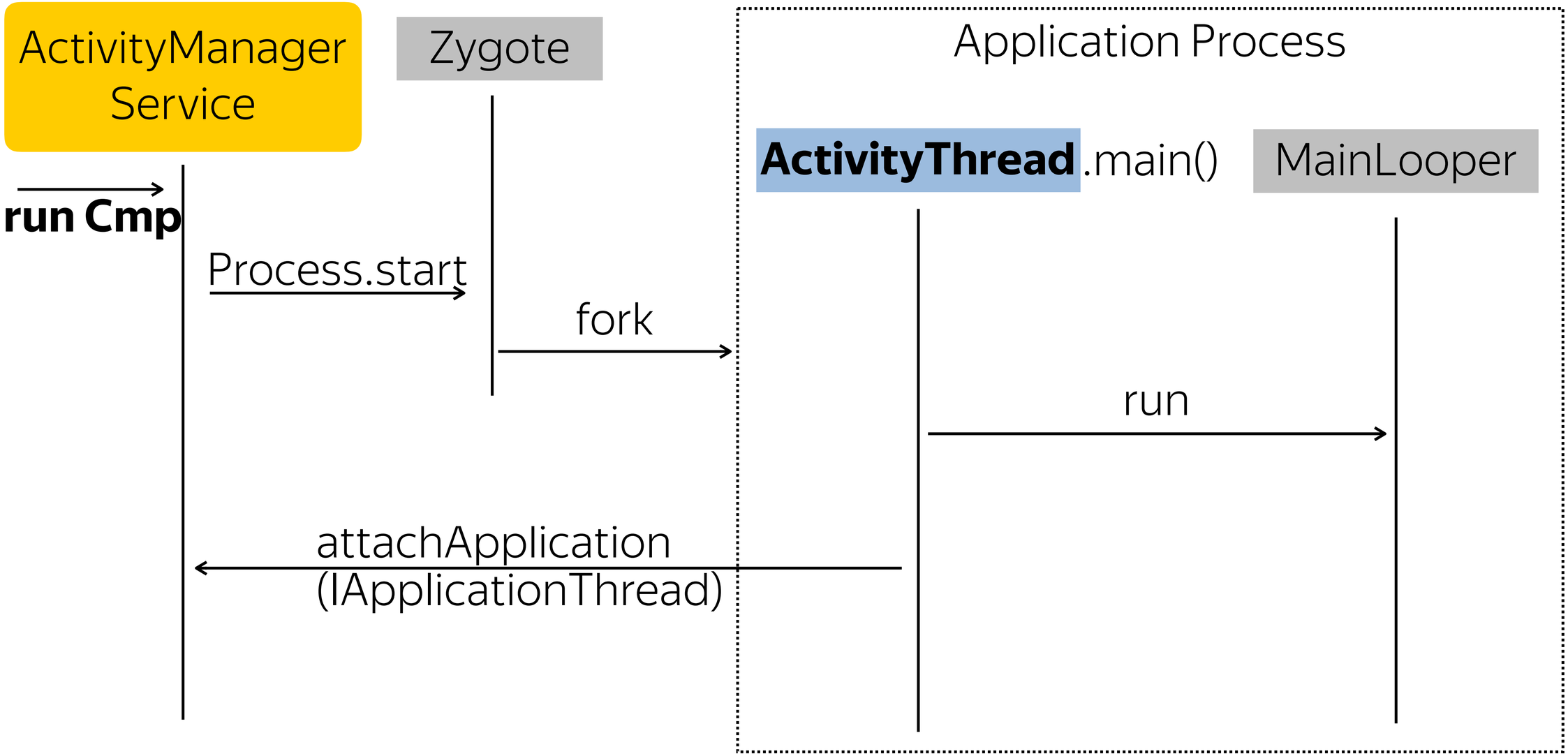
Activity Manager
Power Manager
Package Manager
Account Manager
Alarm Manager
Sensor Service
Window Manager
Status Bar
Location Manager
...

Запуск остальных приложений

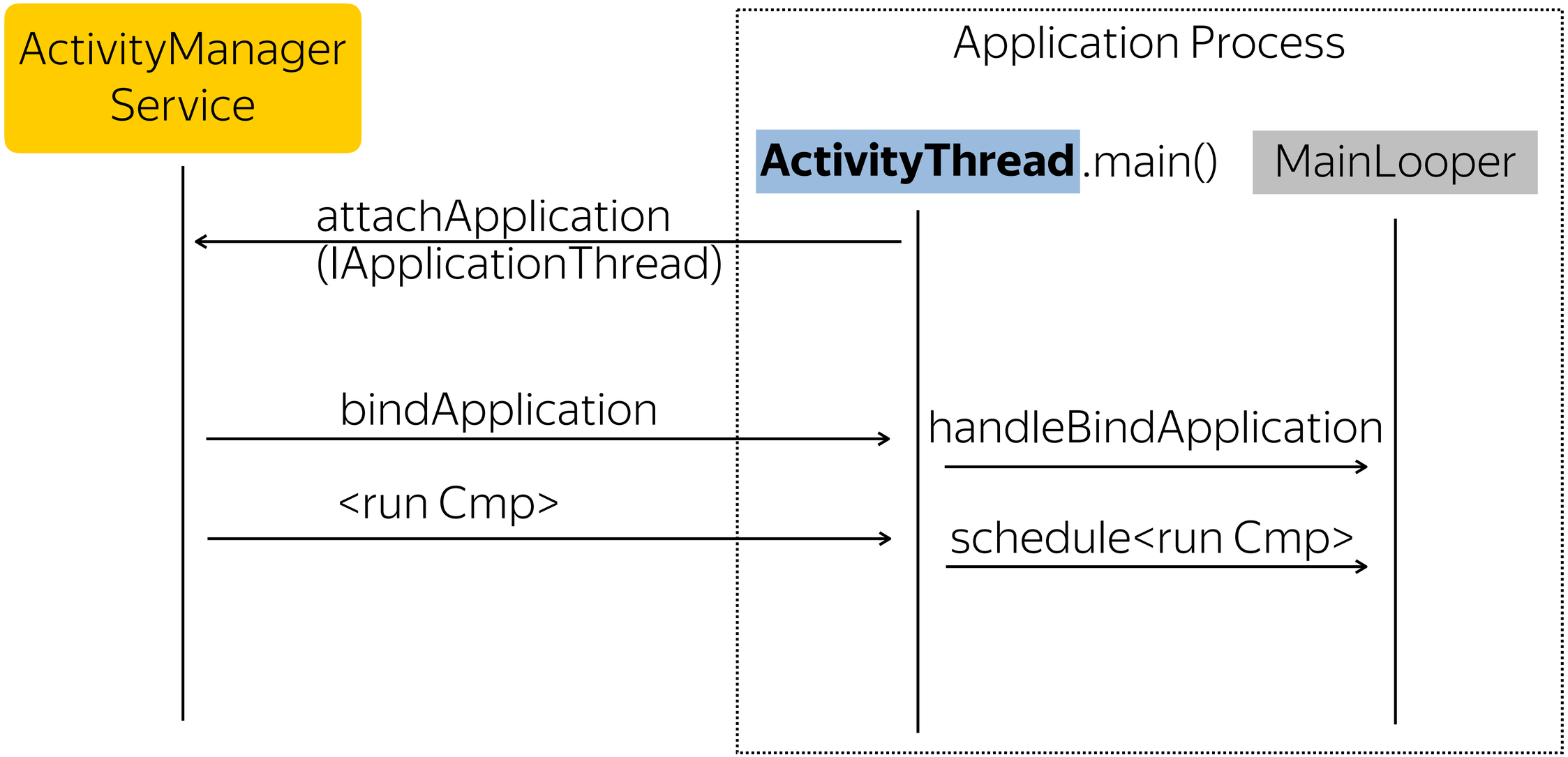
Самый интересный Service



ActivityThread is alive



attachApplication



run<Cmp>

- Запуск Activity, если надо
- Запуск Service, если надо
- Доставка Broadcast

bindApplication

1. Application app = ...makeApplication(...)

2. installContentProviders

- installProvider
- publishProvider

3. app.onCreate()

Как обрабатывается BroacastReceiver

- Если процесса нет, то создать и запланировать через **ActivityThread**
- Если есть, то сразу запланировать через **ActivityThread**
- Если есть в списке зарегистрированных в **ActivityManagerService.mRegisteredReceivers**, то выполнить сразу через **IIntentReceiver.Stub.performReceive**

А как startService & bindService?

ещё проще

- Если процесса нет, то создать и запланировать через **ActivityThread**
- Если есть, то сразу запланировать через **ActivityThread**
- А для bind в ActivityThread ещё возвращается binder

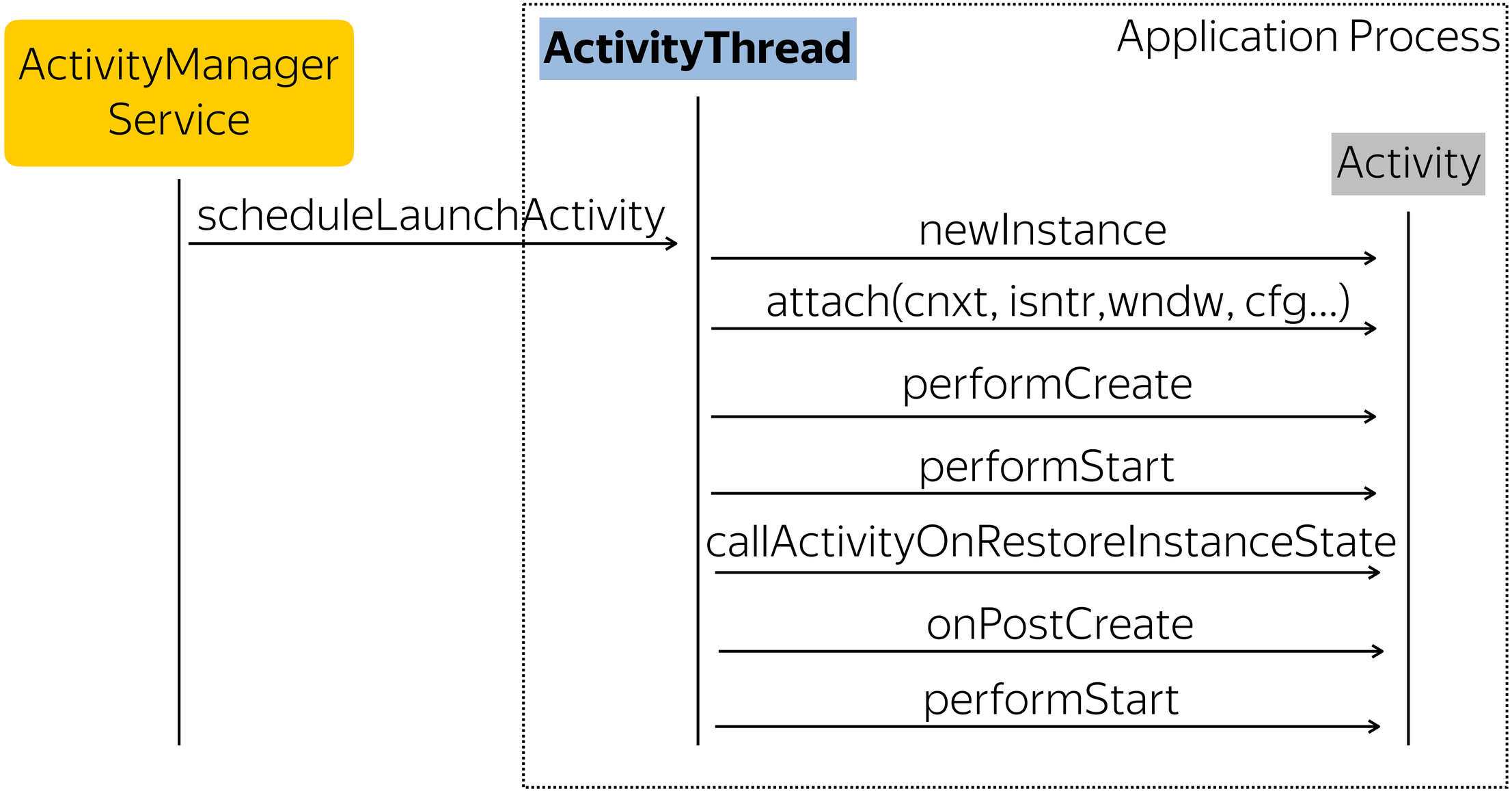
А ContentProvider ?

Всё происходит в
ActivityManagerService.getContentProviderImpl(...)

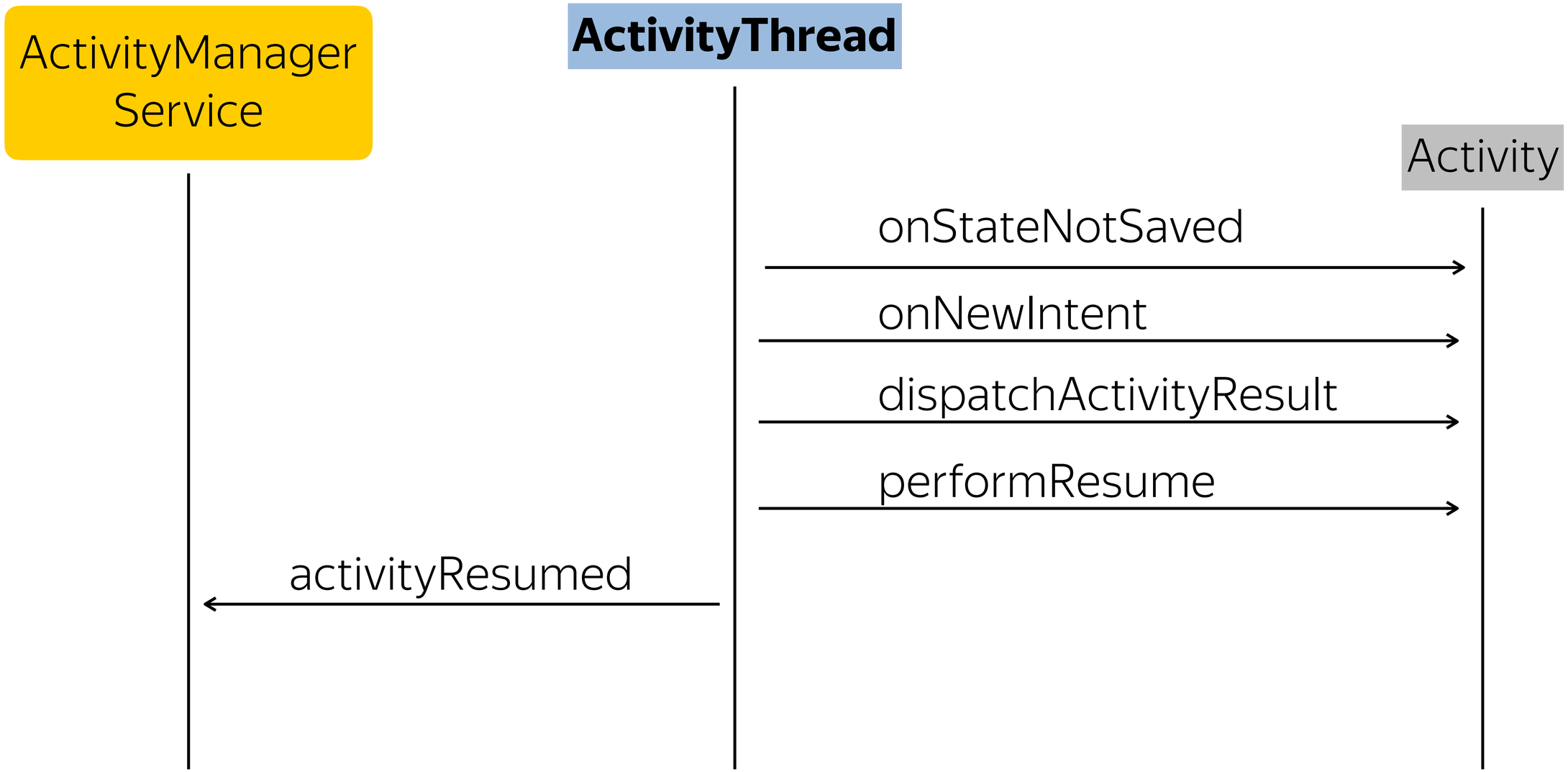
- Если процесса нет, то запустить и ждать когда он опубликует провайдер, а потом его вернуть

- Если процесс есть, то провайдер уже опубликован, а значит он сразу возвращается

А как же Activity lifecycle?



А как же Activity lifecycle?

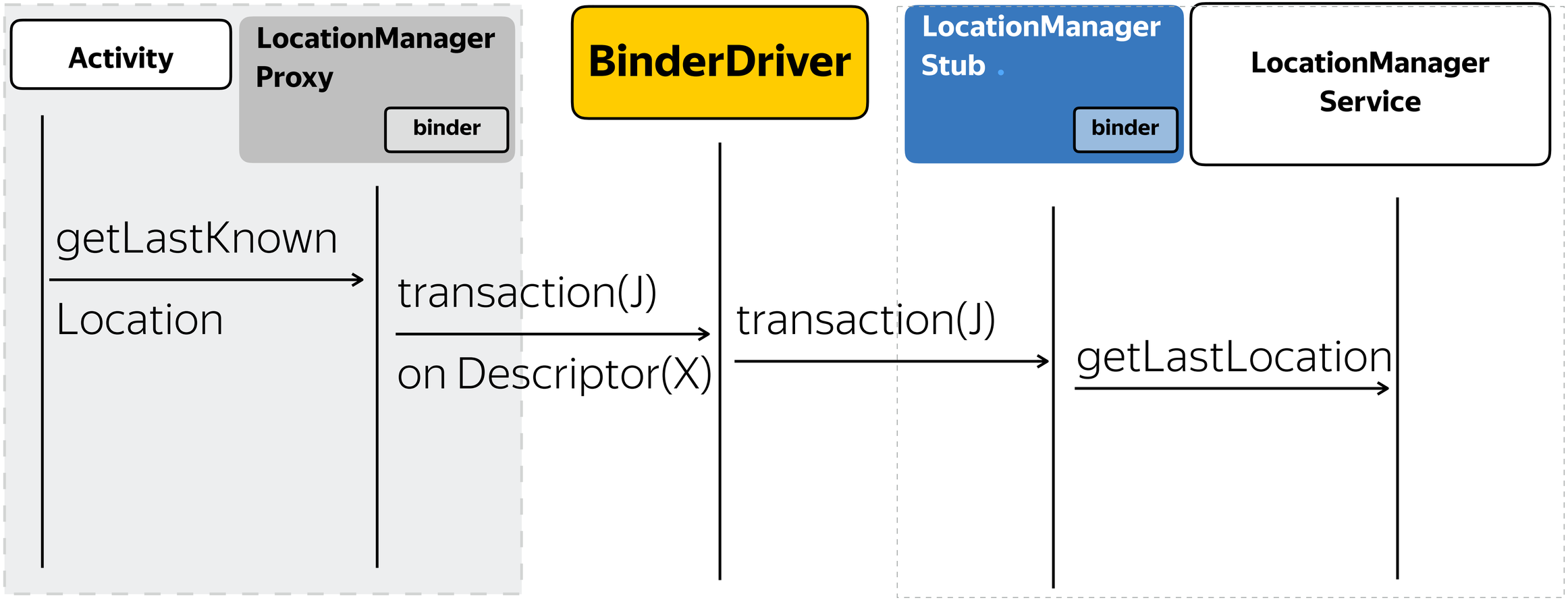


А что потом?

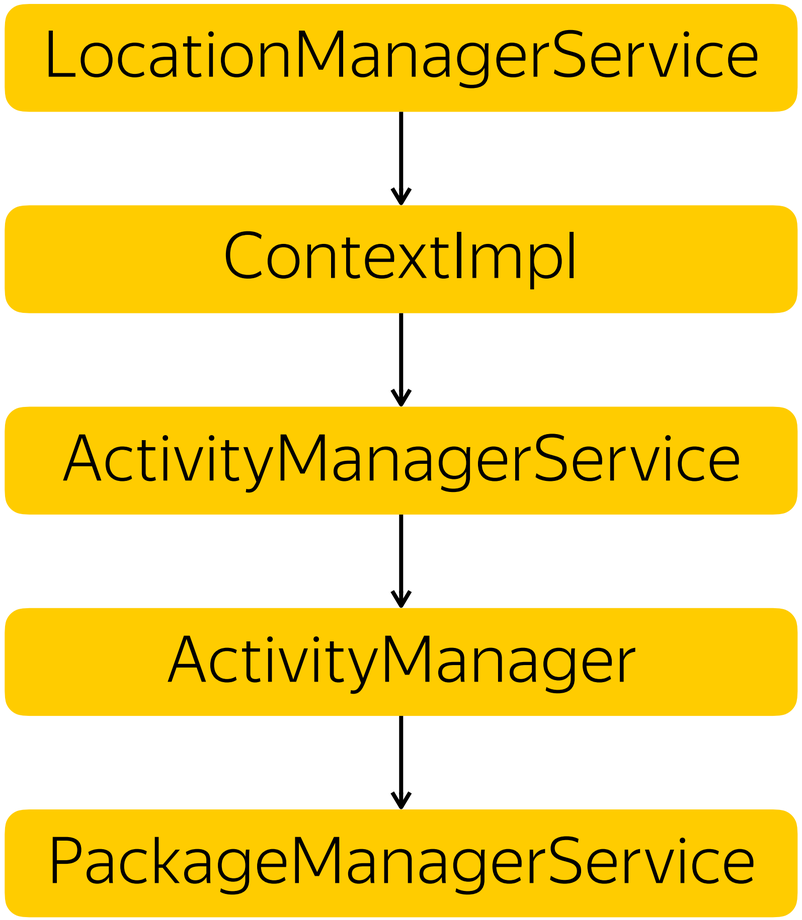
- Взаимодействие с системой
- Взаимодействие с другими приложениями (обычно тоже через систему)
- Потребление ресурсов
- Необработка исключений...

Взаимодействие с системой

Взаимодействие процессов

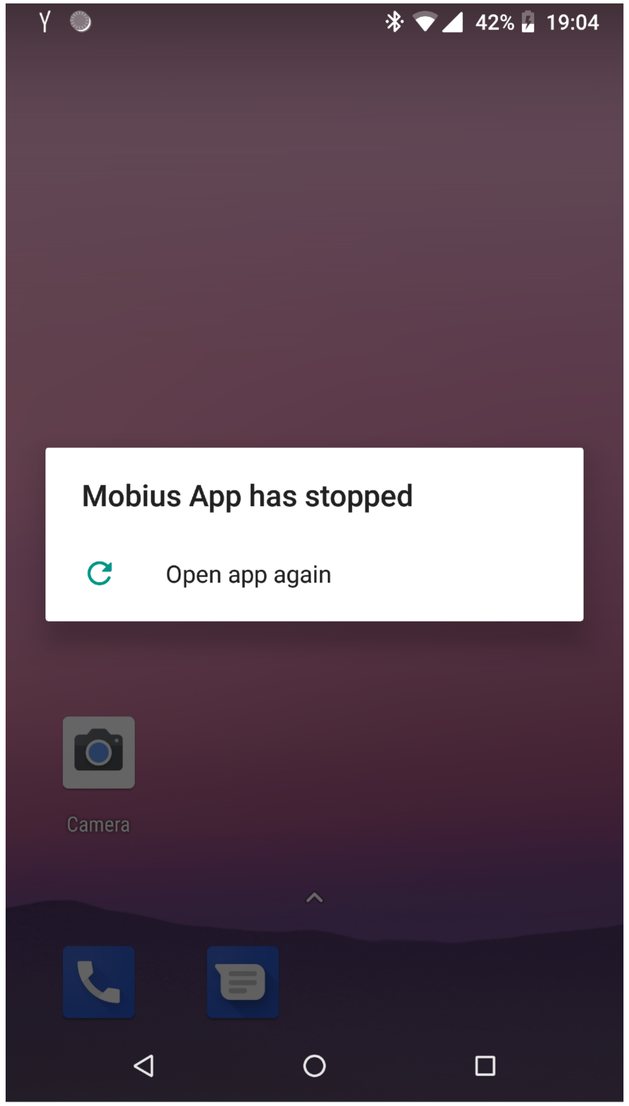


А есть ли доступ?

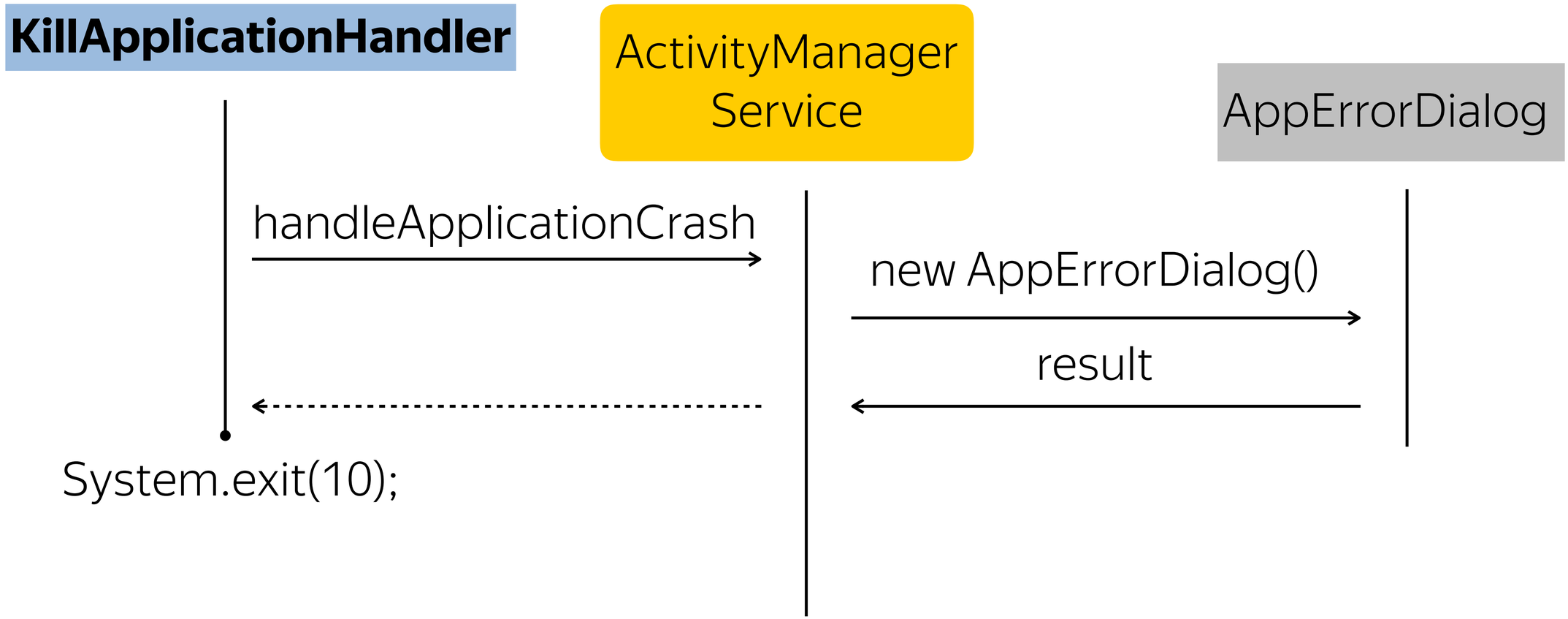


Завершение работы

Что может пойти не так?



При старте процесса у него есть KillApplicationHandler



Свой KillApplicationHandler

```
Thread.setDefaultUncaughtExceptionHandler(new
  Thread.UncaughtExceptionHandler() {
    @Override

    void uncaughtException(Thread t, Throwable e) {
        System.exit(0);
    }

});
```

Сообщи предыдущему ExceptionHandler’у

```
Thread.UncaughtExceptionHandler
excHlr = Thread.getDefaultUncaughtExceptionHandler();

//...

@Override
public void uncaughtException(Thread t, Throwable e) {
    myHandle(t, e);
    excHlr.uncaughtException(t, e);
}

//...
```

Как система решает кого убить первым ...

1. Cached process

- Activities stopped
- No background services

2. Service process

- Service in background

...а кого в последнюю очередь

3. Visible process

- Foreground service
- Paused activity

4. Foreground process

- Resumed activity
- Broadcast receiver onReceive callback
- Service callbacks (onCreate, onStart, onDestroy)

Приоритет зависимости

Про убийство другими приложениями

stopService/unbindService

- Если коротко – нет
- Перемещение в кешированные

Самоубийство

Process.killProcess(Process.myPid())
и
System.exit()

- Полное завершение процесса
- Никакие методы жизненного цикла не будут вызваны

От пользователя

Swipe to close и Clear All

- ActivityManagerService.removeTask()
- корректное завершение Activity
- Процесс будет остановлен, но Sticky Service будет перезапущен через короткое время (если не падал, то 1000ms)

Force Stop

- Kill the process & stop all Services
- Sticky Service перезапущен не будет

Возрождение

Возрождение

STICKY service, почти всегда, будет перезапущен

Ваше приложение сможет работать как раньше,

но
два краша подряд – Ban на доставку ManifestBroadcasts.

Спасибо. Вопросы?

Антон Дудаков
андроид разработчик

bwdude@yandex-team.ru
bwdude
bwdude

Яндекс Авто