Android Performance :UI

学部4年 システムソフトウェア研究室 高川雄平

指導教員：松原克弥

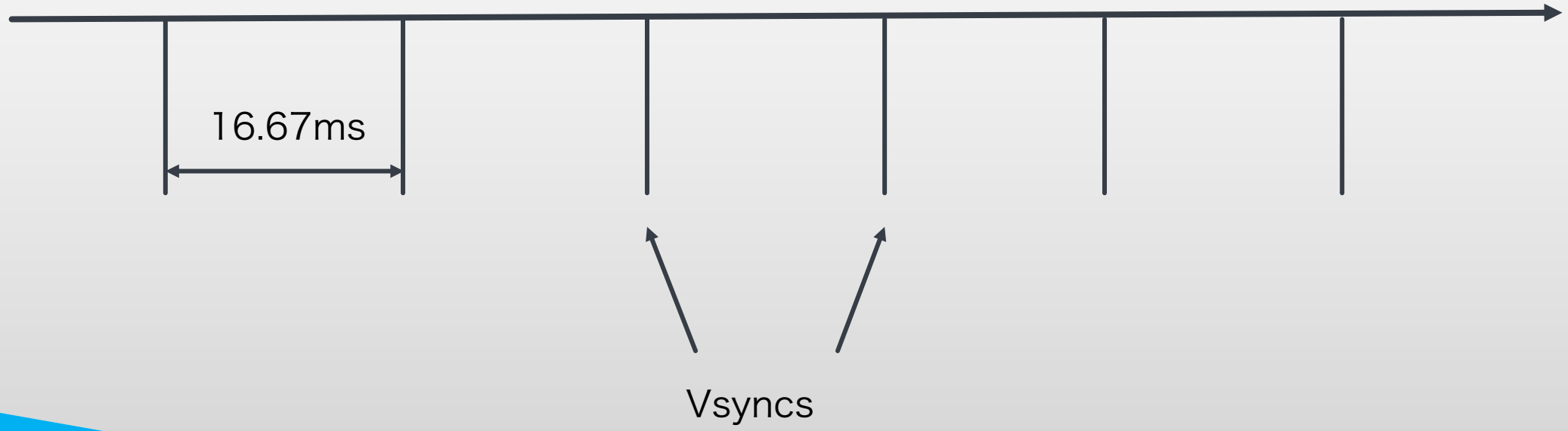
Takagawa Yuhei

UI Performanceに関わること

1. FPS　　：１秒間あたりのフレーム数
2. Frame duration　　：１フレームあたりの秒数
    1. フレームごとにかかる時間が変わる
3. フレーム落ちしない(スキップしない)/一貫性が保たれる
    1. フレームの表示時間の改善

UI Deadlines

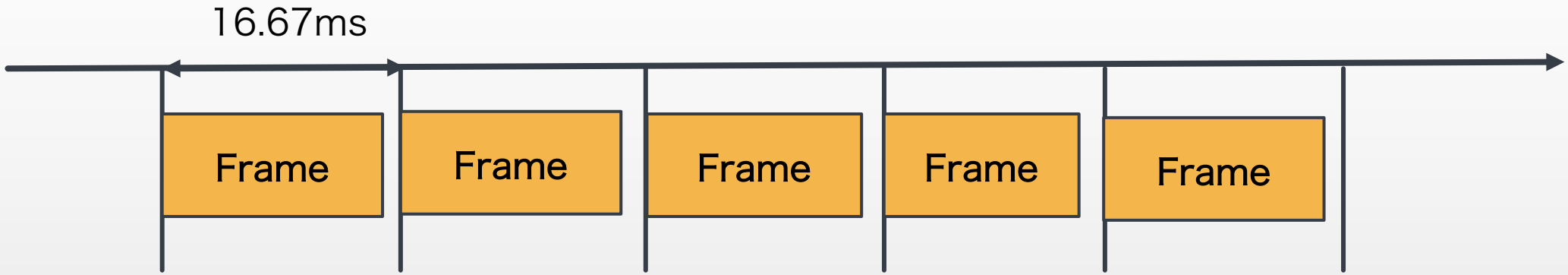
Vsyncs: 画面を上から下まで表示する信号　16.67msの間隔
ユーザはVsyncsによって変更された画面を見る



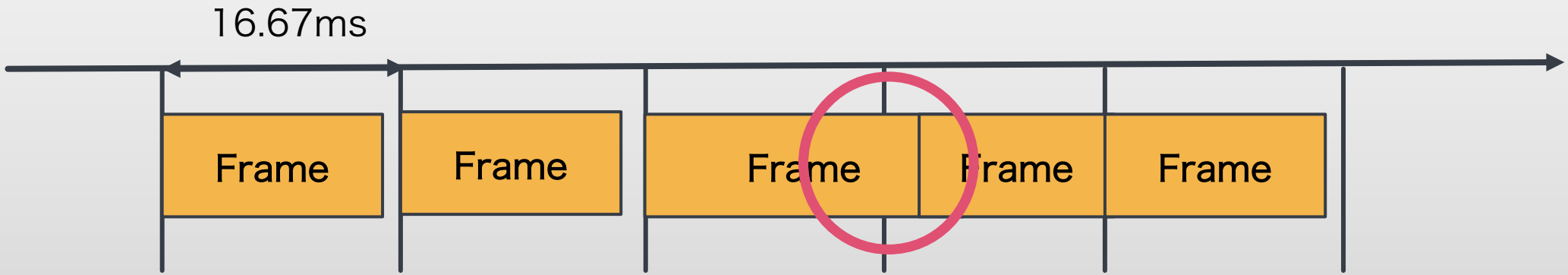
UI Deadlines

Vsyncs: 画面を上から下まで表示する信号　16.67msの間隔

Good



Bad
フレームが
スキップした(落ちた)
ように見える



レンダリング

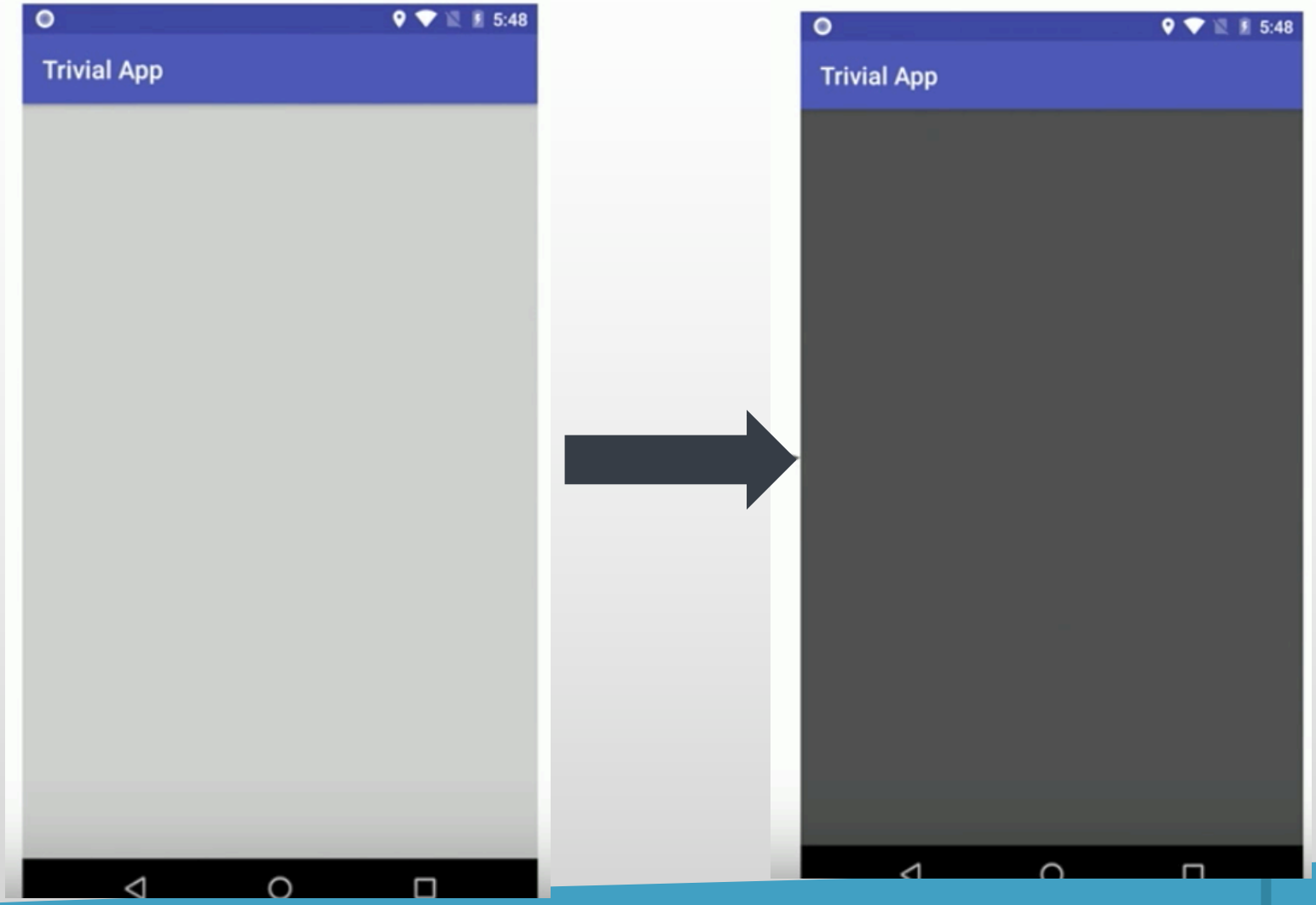
- プログラムを用いて画像や映像などを生成すること
- 画素を作り出す
- 画面を表示する/更新する

レンダリングで何が起きているか – アプリから見たフレーム -

```
@ColorInt
Int mColor = Color.LTGRAY;

@Override
Public void onClick(View v){
        mColor = Color.DKGRAY;
        invalidate();
}

@Override
Protected void onDraw(Canvas canvas){
        canvas.drawColor(mColor);
}
```



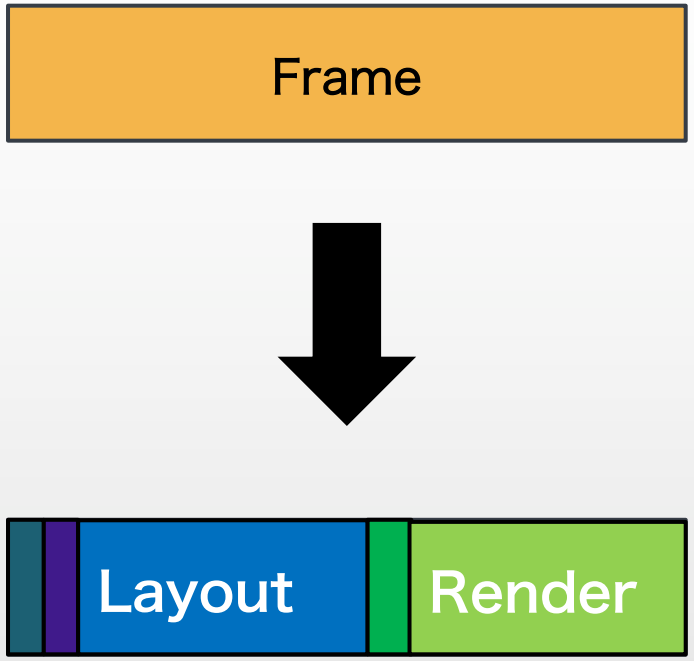
invalidate()で前の画面描写がクリアされる

onDraw()が呼び出され画面が再び描写される

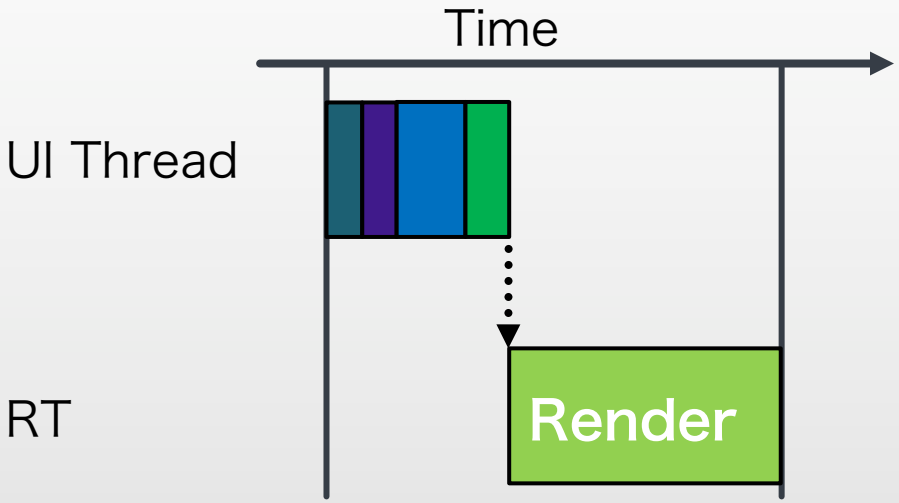
invalidate()はそのままonDraw()を呼び出さない
→呼び出しまでの時間がかかる

レンダリングで何が起きているか - プラットフォームから見たフレーム -

1. Input 入力
2. Animation 演出
3. Layout 再描写のための変更
4. Record Draw 変更の記憶
5. Render 画面の描写



レンダリングで何が起きているか – フレームのためのスレッド -



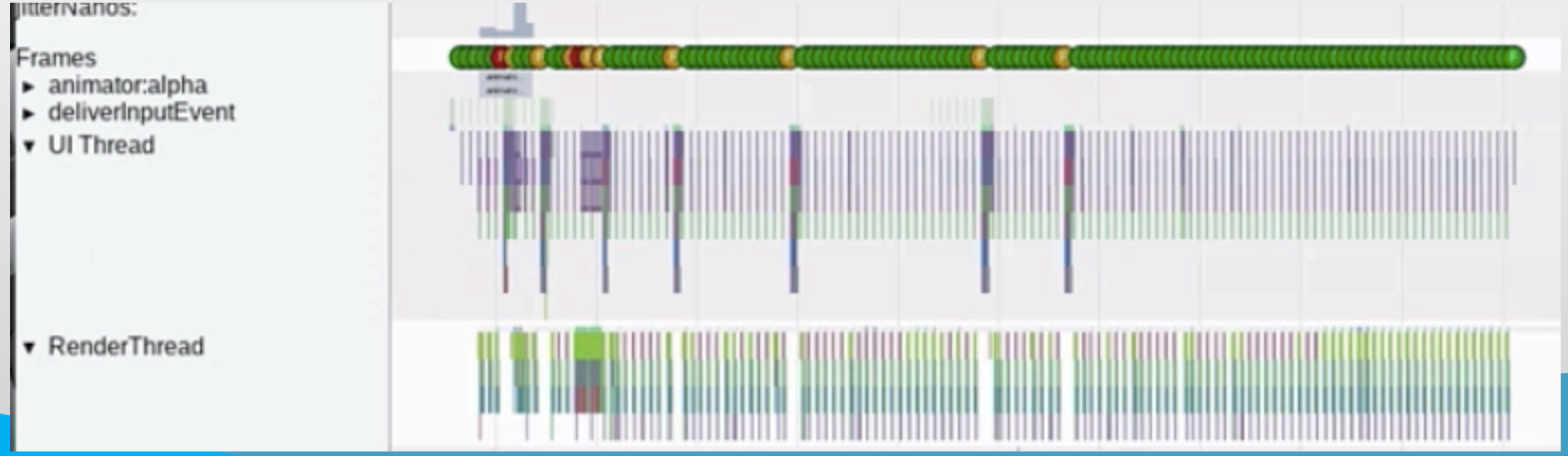
UI Thread : UIのための処理を行うスレッド

RT(Render Thread)：レンダリングをするためのスレッド

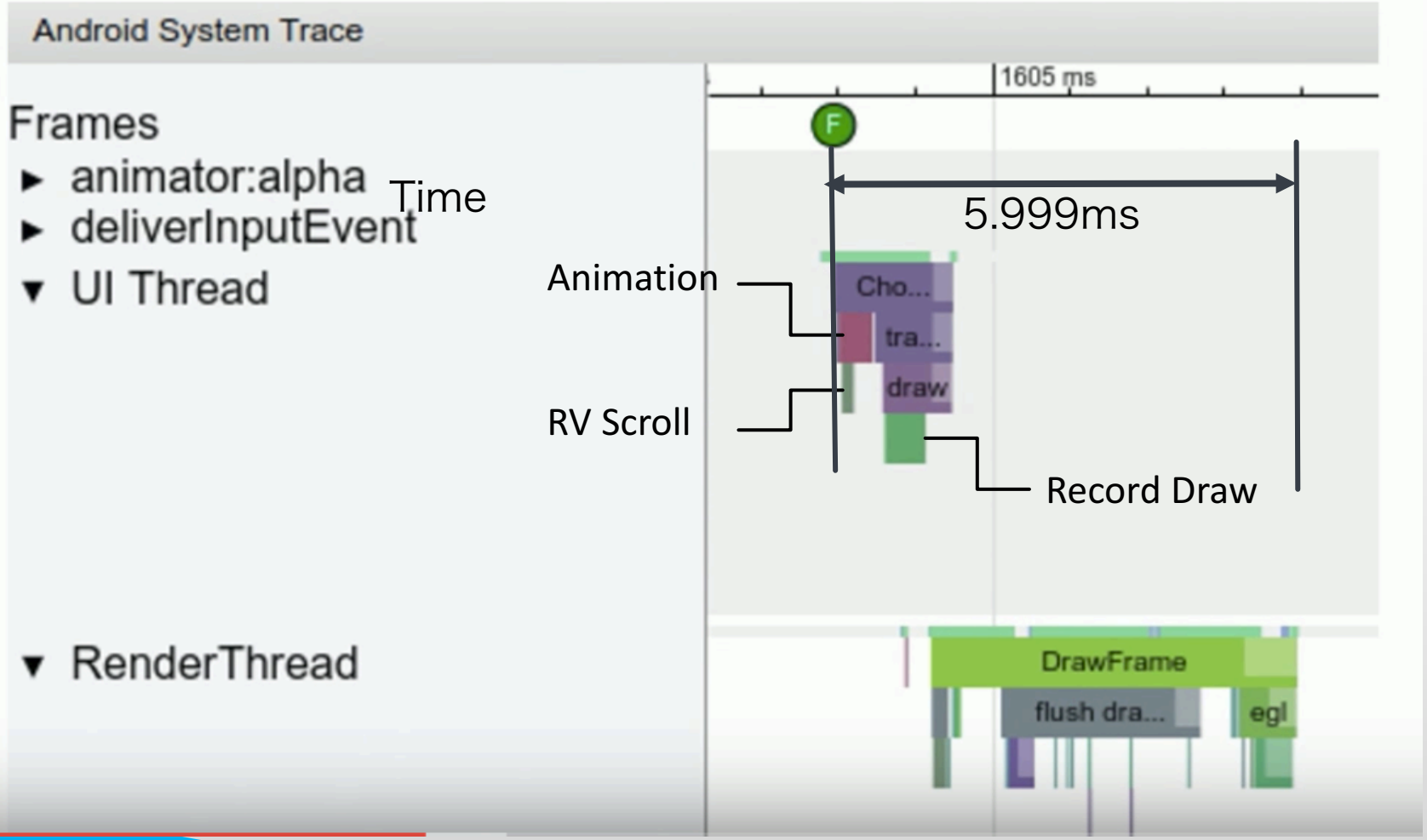
レンダリングで何が起きているか – 詳細的なフレーム -

Jankへの対処方法

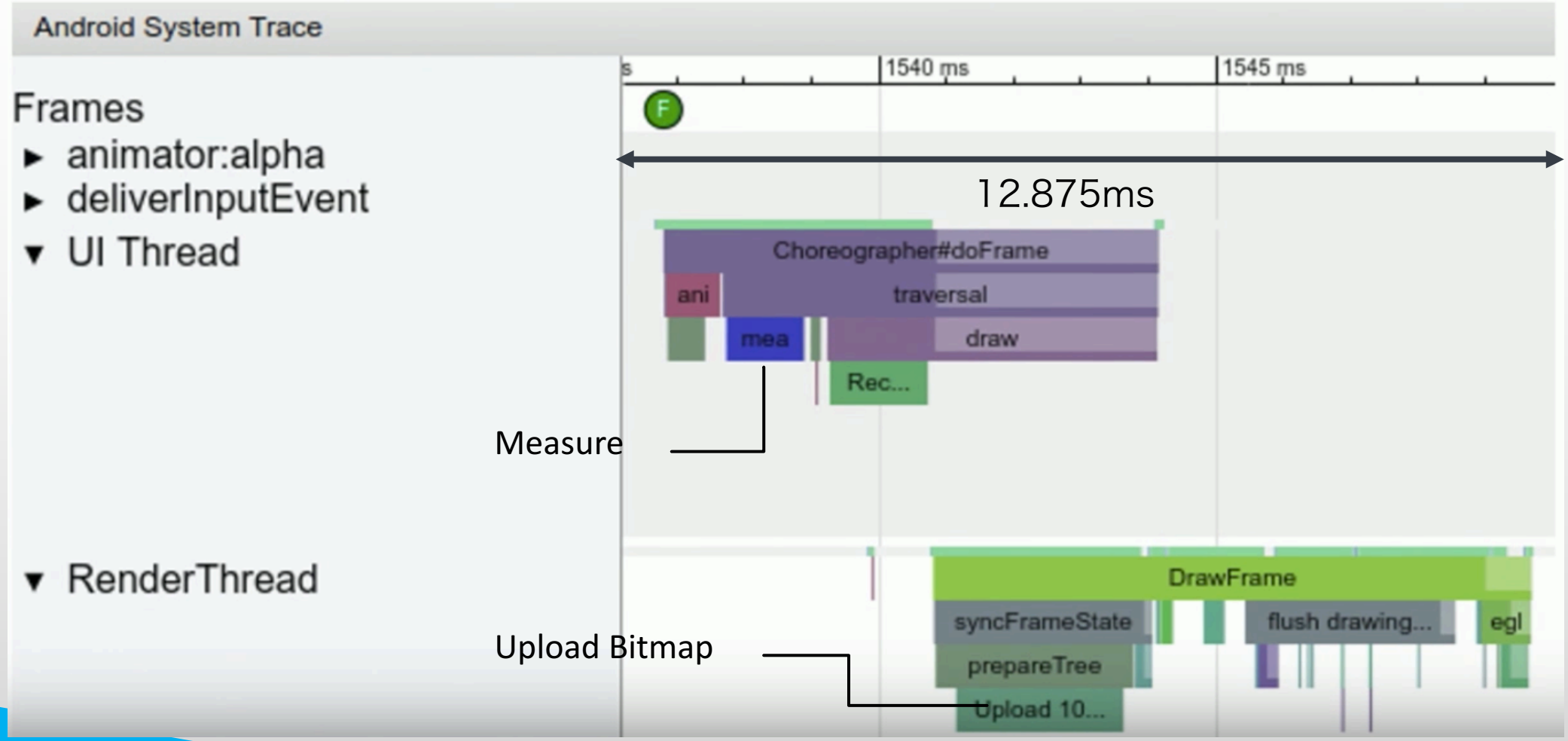
1. (release)ish buildをして実行する
    1. 実行時にはデバッグをoffにする onにすると余計なデータが混じることもある
2. アプリのレンダリングに関するものをグラフ化する
    1. “Profile GPU Rendering”を使えばできる
3. Systraceを使う
    1. Androidのパフォーマンス解析ツール



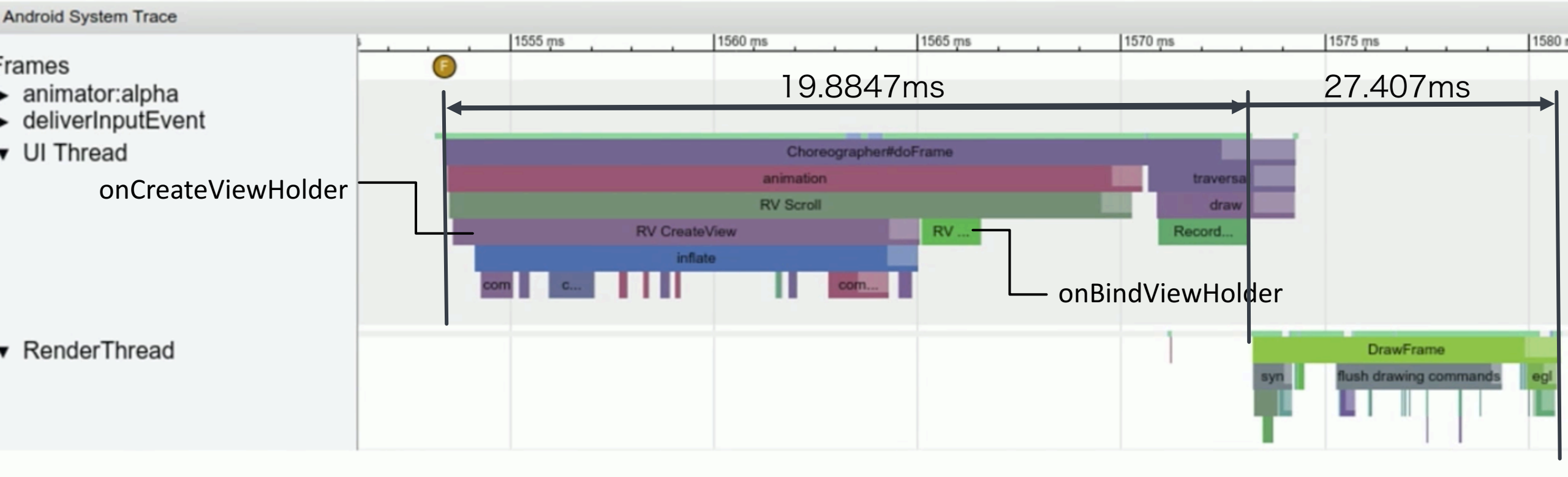
Systraceの例 - 簡単なフレーム RecyclerView good



Systraceの例 - 簡単なフレーム RecyclerView normal

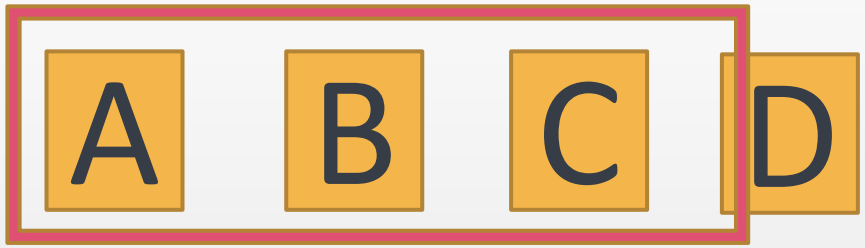


Systraceの例 - 簡単なフレーム RecyclerView bad



RecyclerView のPrefetch機能

アイテムを事前に読み込んでおくことができる



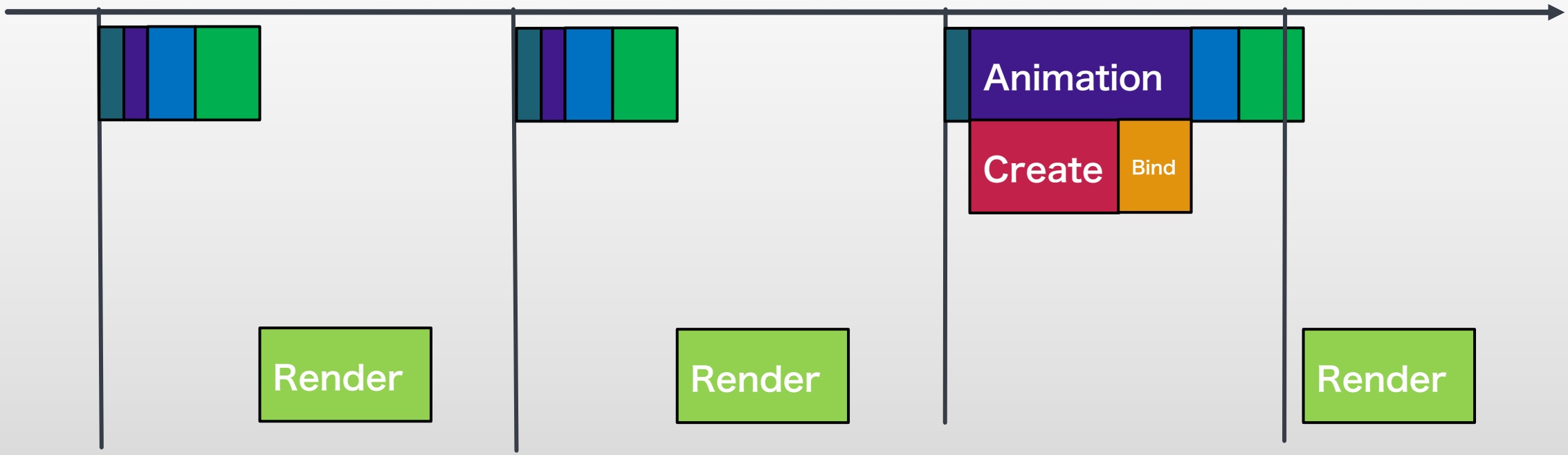
Scroll↓

DはPrefetchで事前に
読み込まれていた

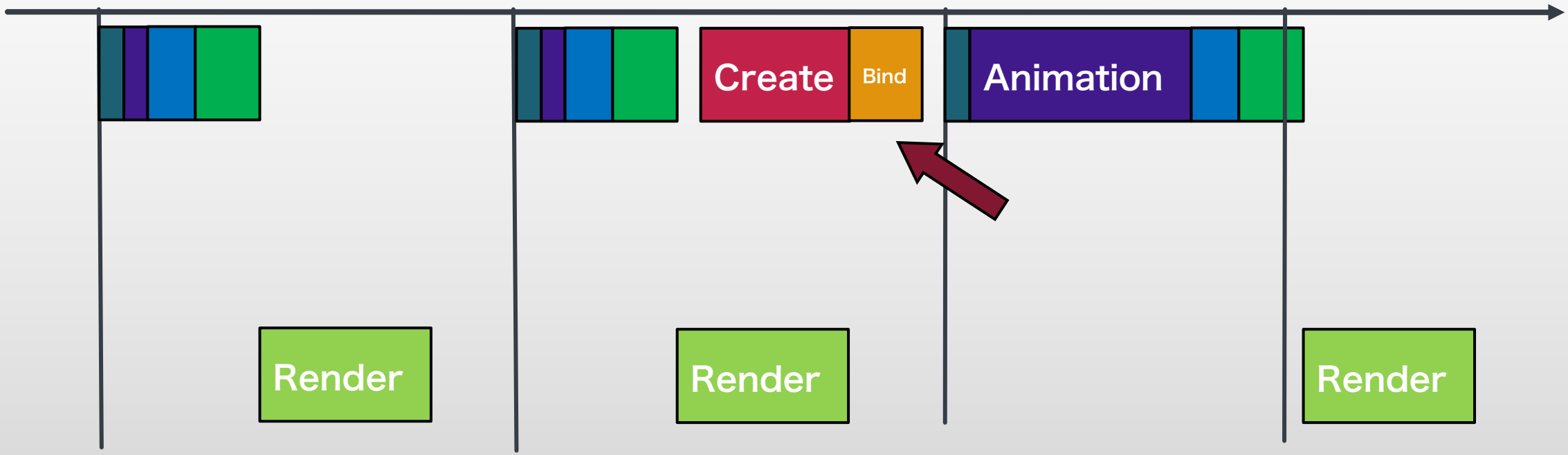
Scroll↓

EはPrefetchされていないので
読み込む処理が必要

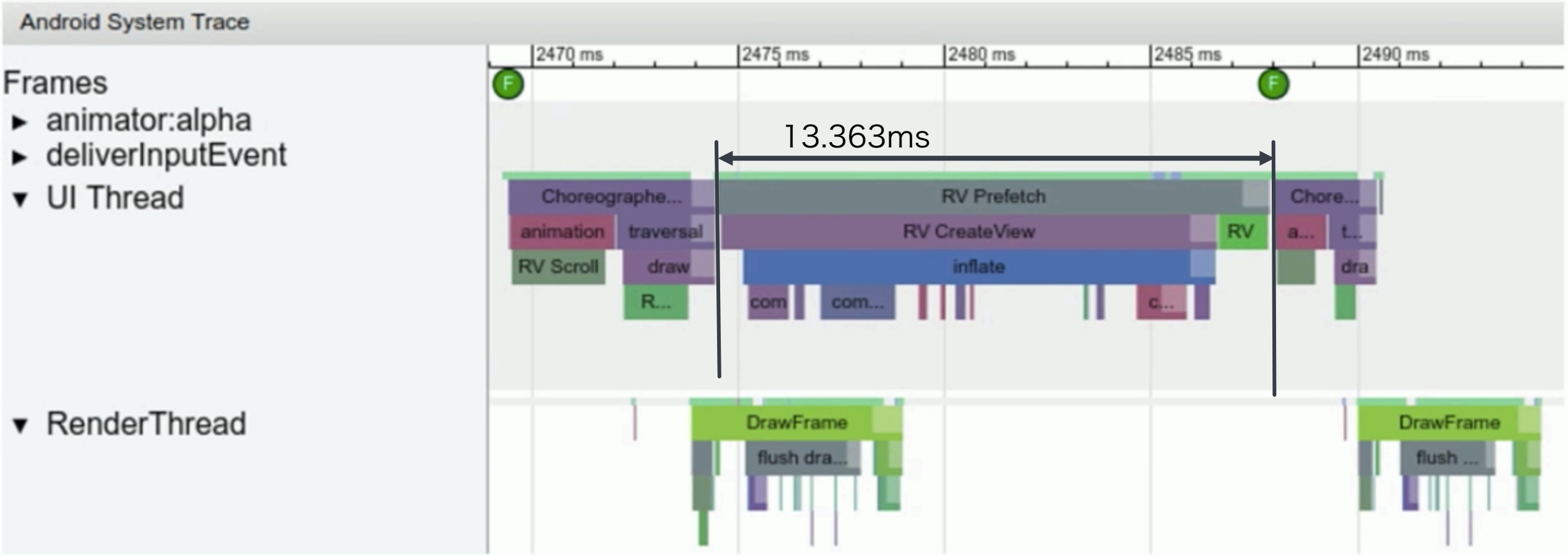
RecyclerViewでのCreateViewの過程



RecyclerViewでのCreateViewの過程 (prefetch)



Systraceの例 - 簡単なフレーム RecyclerView bad



Jankの原因

- Garbage Collection(GC)
  - プログラムは大きくならないが，Runtimeの改善は必要
- OSスレッドのスケジューリング
- Rendering
  - View の使い方
  - Bitmapの送信
- View Recycling
  - Binderをたくさんしていたりする

アプリのパフォーマンスを維持するには

- ユーザ情報を収集する
- 原因を追跡する
- Jankのテストをする
- ツールを使う

- d.android.com/vitals
- Android Vitalsを使うと便利です
- Android stadio も結構便利です