Quantify and Optimize User Interactions with Android Devices

Xiao-Feng Li
xiaofeng.li@gmail.com

OCT, 2011

Summary

- Performance is not enough to characterize a client device
- An systematic approach has been established to engineering user interactions
  - AWS is a comprehensive engineering workload suite
  - UXtune is a toolkit to assist analysis and optimization
- Android user interactions can be impacted by the technical factors across the whole software stack

User interaction is a new challenge but not insurmountable

Agenda

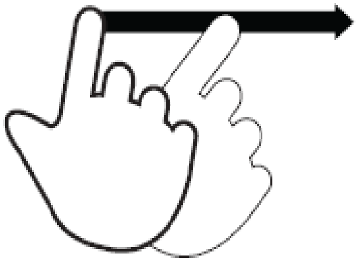
- Client device user interactions
- Android user interaction optimizations
  - Android Workload Suite and Android UXtune
- Case studies of Android optimizations
- Factors that impact Android user interactions
- Summary
- Information

Performance Is NOT Enough

- Performance does not reflect consistently as user perception
- Performance is only about the behavior of system in *steady state*
- Then, what are missing?

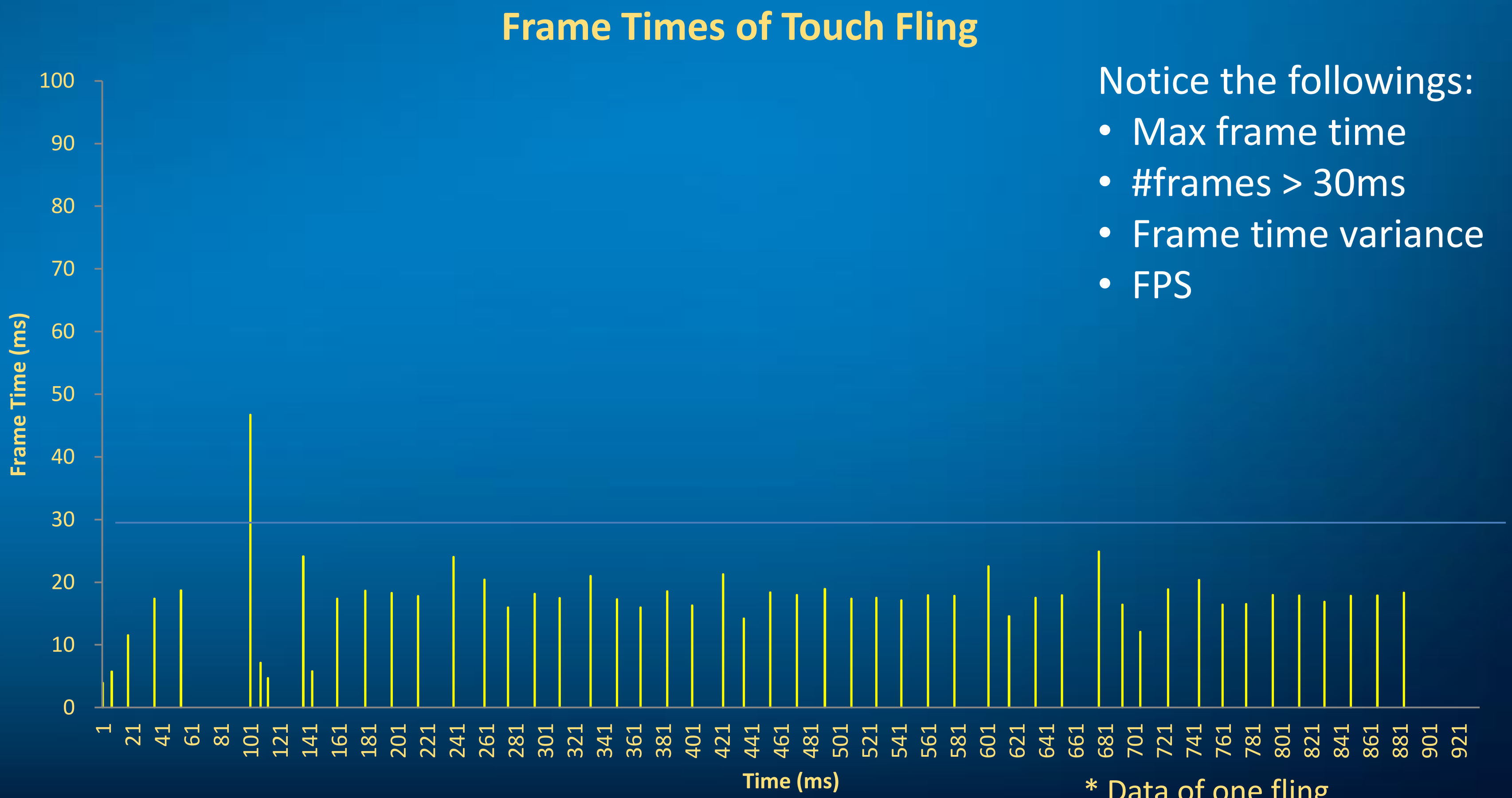
Observation of Touch Fling Operation

- Fling the picture in Android Gallery application
  - Current picture slides and switches to next picture



Quickly brush surface with fingertip

Frames of A Fling Process on Device A



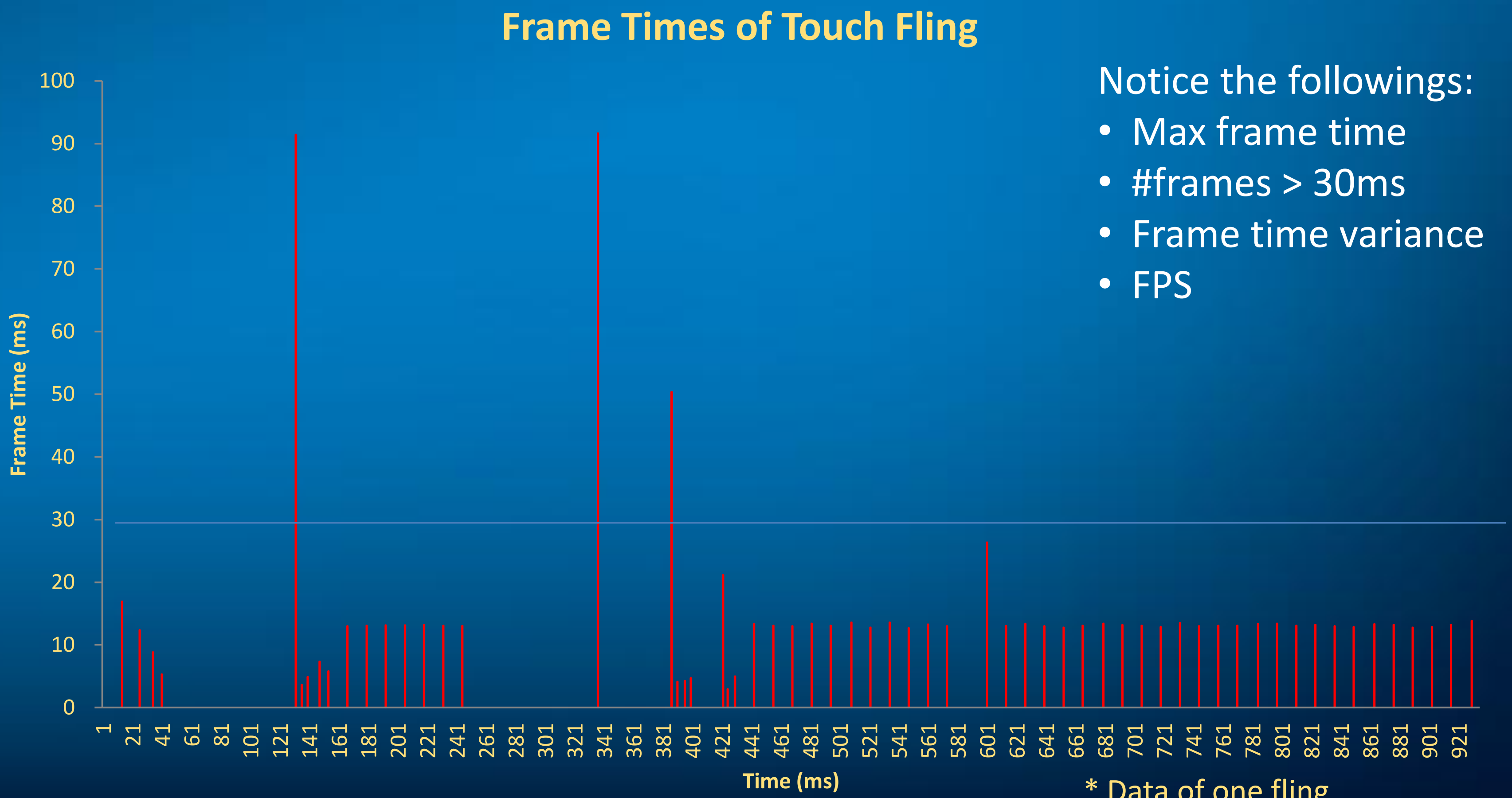
Notice the followings:

- Max frame time
- #frames > 30ms
- Frame time variance
- FPS

* Data of one fling

2011-09-12

Frames of A Fling on Device B (Higher FPS)

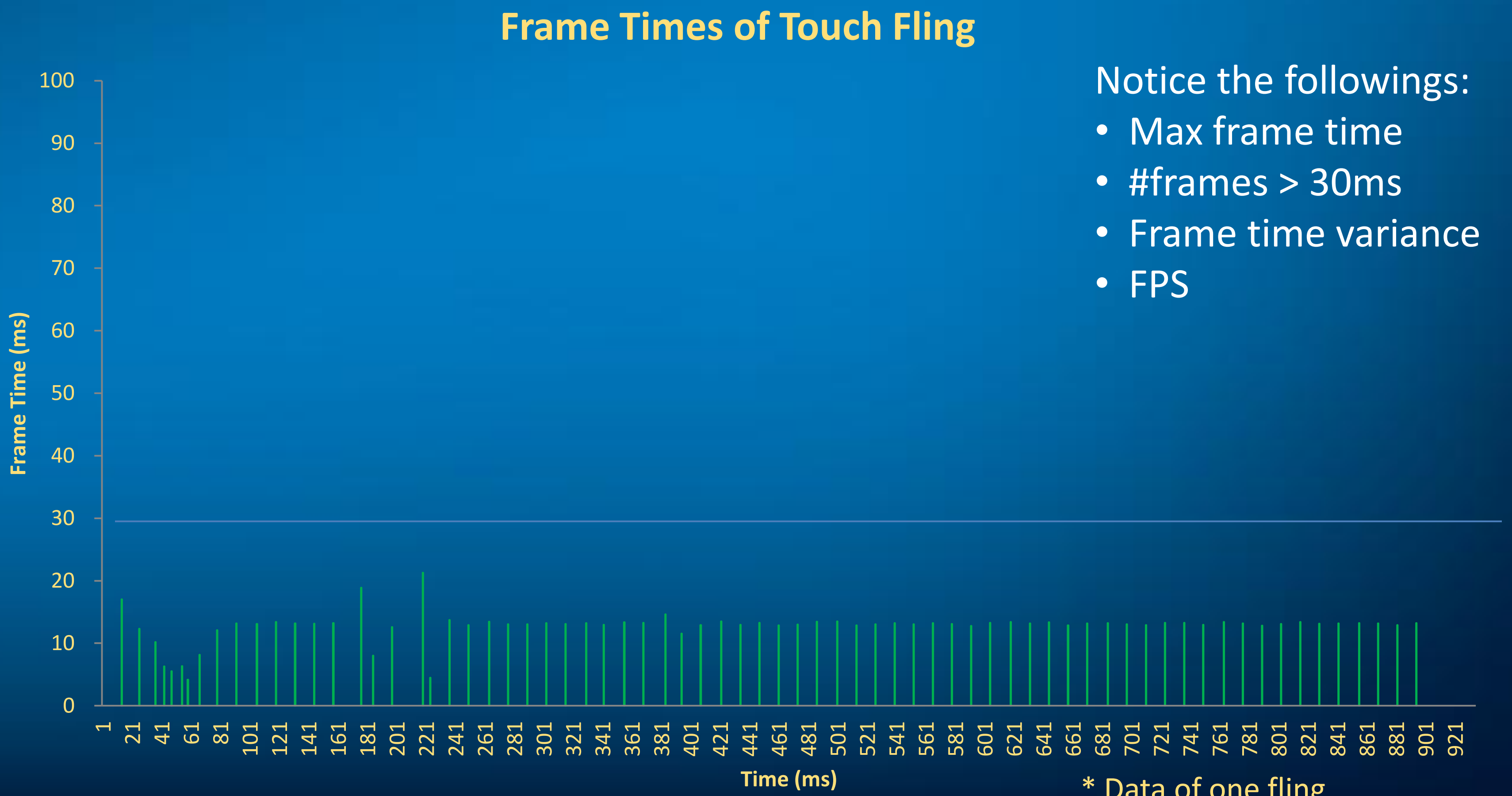


Notice the followings:

- Max frame time
- #frames > 30ms
- Frame time variance
- FPS

* Data of one fling

A Fling on Device B After Optimization



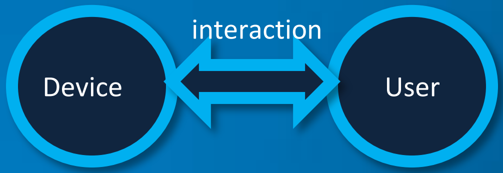
Notice the followings:

- Max frame time
- #frames > 30ms
- Frame time variance
- FPS

* Data of one fling

User Interactions with Client Device

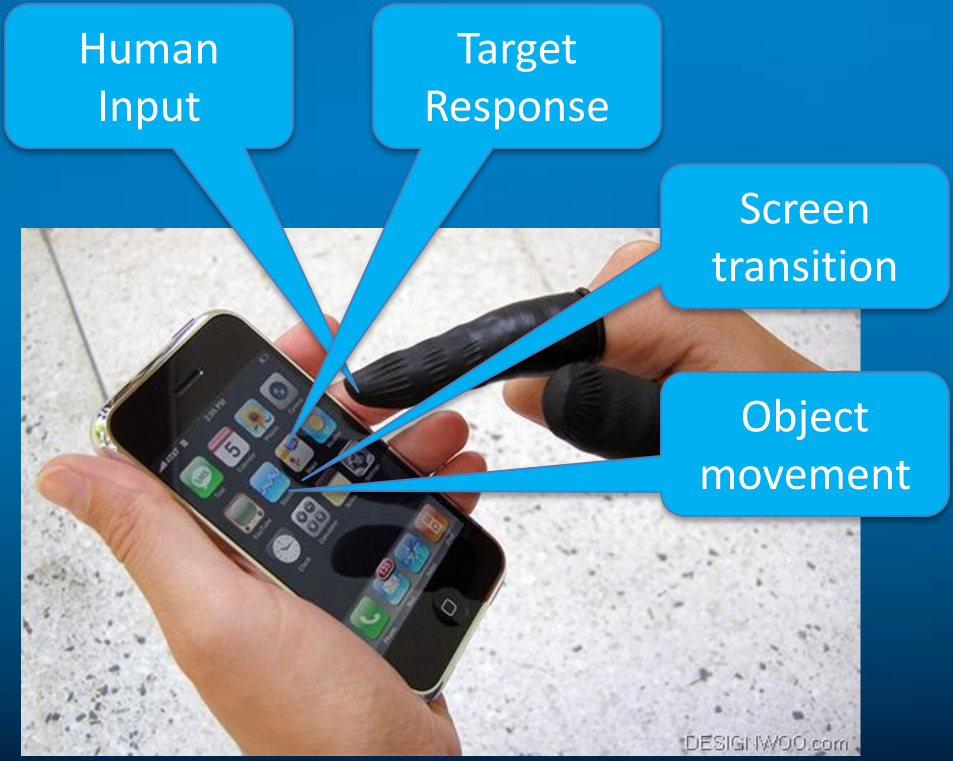
- A sequence of interactions



Device

interaction

User



Human Input

Target Response

Screen transition

Object movement

- Does the input trigger the target **correctly**?
- Does the system act **responsively**?
- Does the graphics transition **smoothly**?
- Does the object move **coherently**?

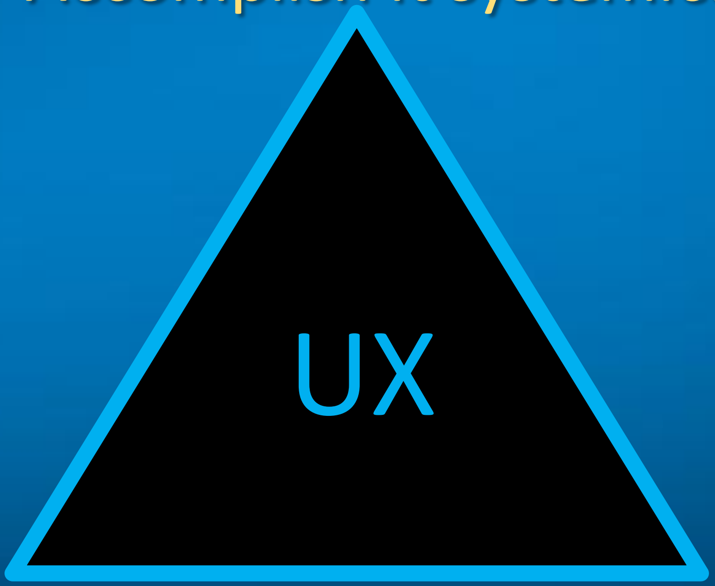
User Interactions in Software Engineering

- *Dynamic state transitions* in the software stack
  - Traditional performance more about *steady state*
  - Performance is the links of the interaction chain
- User Interaction evaluation is a superset of traditional performance evaluation
- For example: a video workload
  - *"launch player"* → *"start playing"* → *"seek progress"* → *"video playback"* → *"back to home screen"*
  - Traditionally, only "*video playback*" is evaluated

A Bit on User Experience Philosophy

- UX is not an objective process, but an interactive process with subjective factors
  - Consider watching movie or listening to music
- No silver-bullet to measure UX
  - Current academic research status with eye-tracking, heart-beat, poll, etc.

UX Philosophy Triangle



Logos:
“Accomplish it systemically”

Ethos:
“Do it correctly”

Pathos:
“Follow it naturally”

Agenda

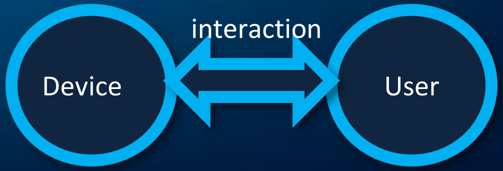
- Client device user interactions
- Android user interaction optimizations
  - Android Workload Suite and Android UXtune
- Case studies of Android optimizations
- Factors that impact Android user interactions
- Summary
- Information

Interaction Measurement Criteria

- Measure the critical path of user interactions in software stack
- Criteria
  - Perceivable (by a human being)
  - Measureable (by different teams)
  - Repeatable (in multiple measurements)
  - Comparable (between different measured systems)
  - Reasonable (about the causality)
  - Verifiable (for an optimization)
  - Automatable (largely unattended, not strictly)

Interaction Measurement Aspects

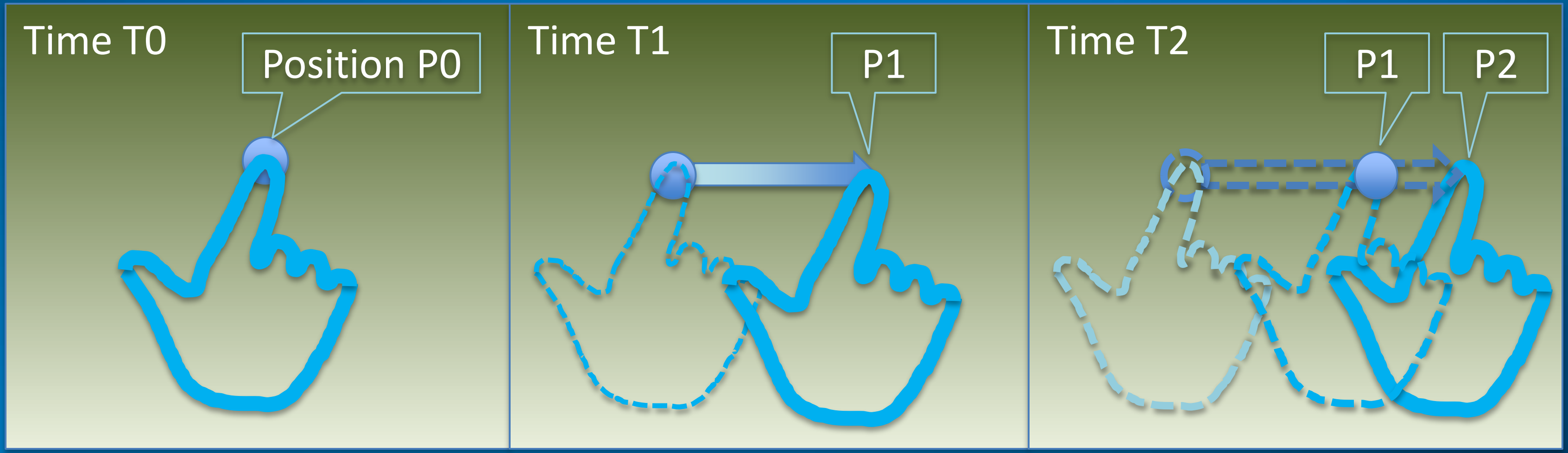
- User controls device (subject → object)
  1. **Accuracy/fuzziness**: Range/resolution of inputs that can trigger a correct response
  2. **Coherence**: Object move delay, difference in move trajectory
- Device reacts to user (object → subject)
  3. **Responsiveness**: Time between an input delivered to the device response, and to the action finish
  4. **Smoothness**: Maximal frame time, frame time variance, FPS and frame drop rate



Example Industry Experience Values

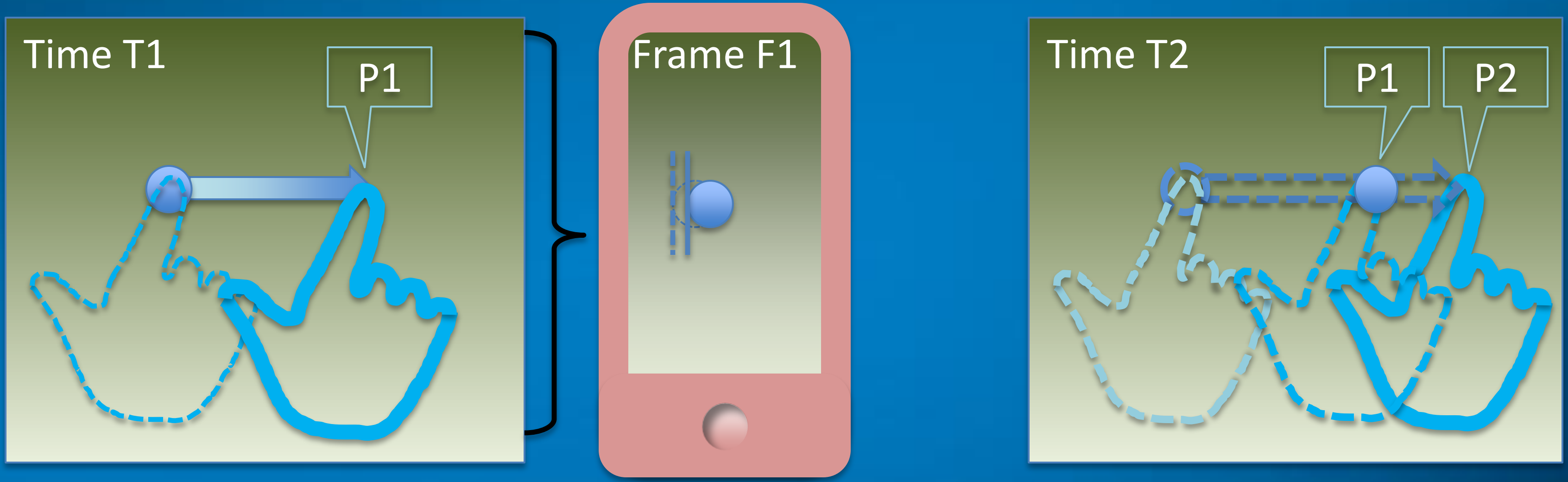
	Best	Good	Acceptable
Response delay	100ms	200ms	500ms
Graphics animation	120fps	60fps	30fps
Video playback	60fps	30fps	20fps

Example "Drag": Drag Icon in Homescreen



- Measure drag coherence
  - P1: the position where the icon starts to move at T1
  - T2: the time when the icon reaches P1
  - P2: the position where the finger touches at time T2
  - P1 - P0, P2 - P1, T2 - T1 are the smaller the better

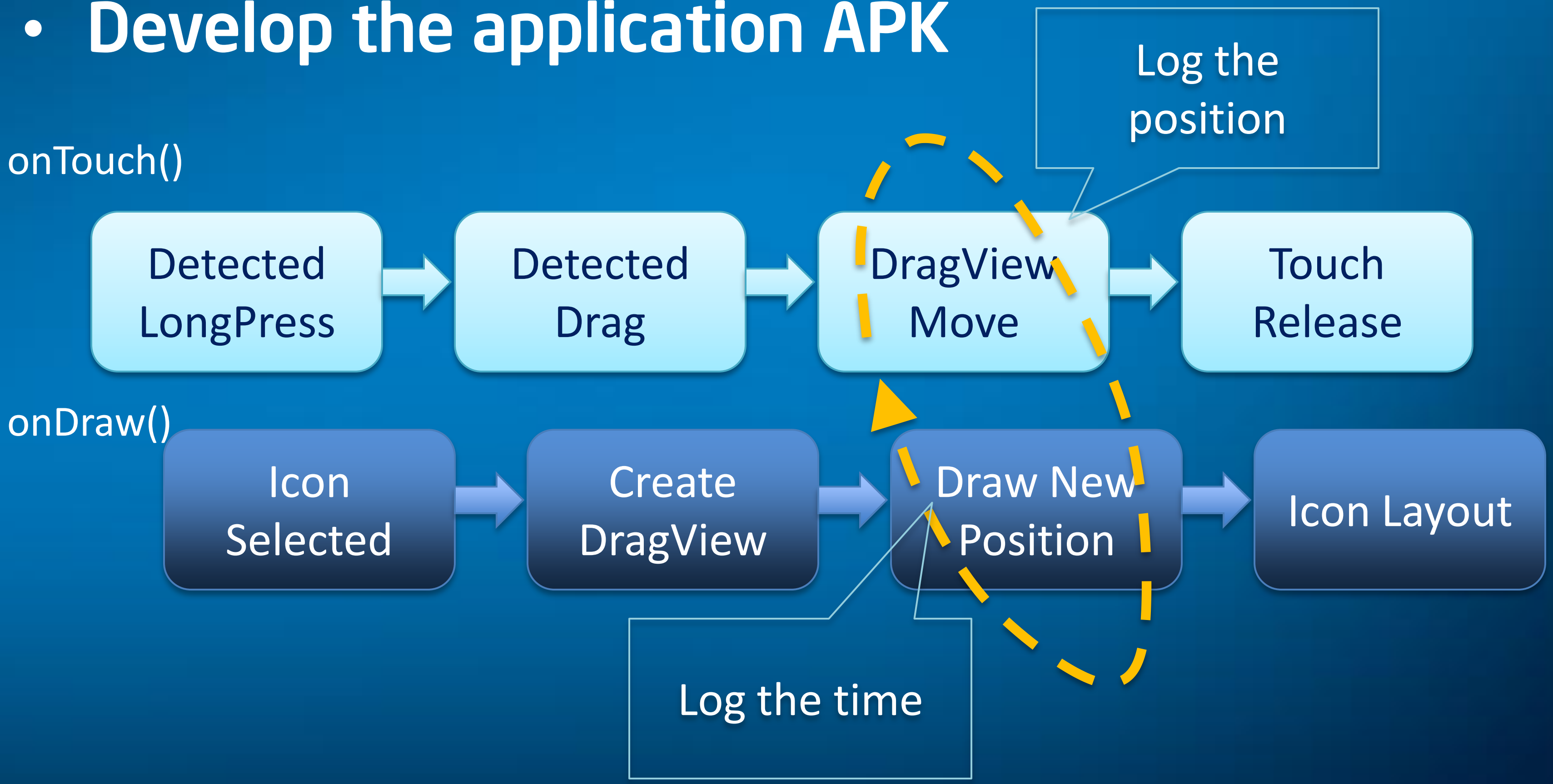
Example "Drag": Map To Engineering Values



- Metrics computation
  - T1 = Time when Frame F1 is drawn by SurfaceFlinger
  - P1 = Position value of the touch event at time T1
  - T2 = Time of the frame when icon's position is P1
  - P2 = Position value of the touch event at time T2

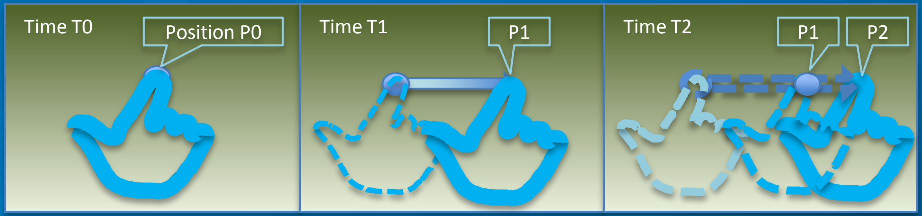
Example "Drag": Dev Engineering Workload

- Develop the application APK



- Note: the touch event time-stamp is not the exact finger touch time. There is a few ms difference. It does not impact the drag lag distance optimization purpose.

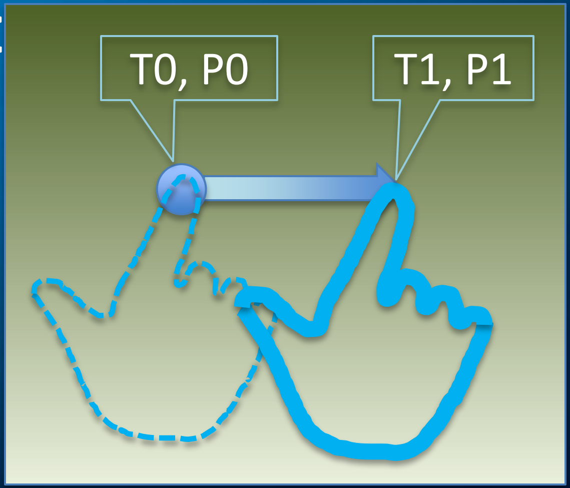
Example "Drag": Optimization Algorithm



- Analysis
  - T0: the event is delivered to system
  - T1: the app finishes drawing the first frame of movement
    - T1 – T0 is the Android processing time, cannot be 0
    - P1 – P0 is the distance finger moves during T1 – T0
- Two complementary optimization approaches
  - Optimize the execution path to reduce T1 – T0
  - Draw the icon at predicted position such as P2

Example "Drag": Predicting Drag Position

- Move icon to the finger position when next frame is drawn
  - $\text{SPEED}_{\text{finger}} = (P1 - P0)/(T1 - T0)$
  - $\text{TIME}_{\text{frame}} = 1/\text{FPS}$
  - $\text{MOVE}_{\text{finger}} = \text{SPEED}_{\text{finger}} * \text{TIME}_{\text{frame}}$
  - $\text{NextPOS}_{\text{finger}} = \text{MOVE}_{\text{finger}} + \text{Current}$
  - $\text{NextPOS}_{\text{icon}} = \text{NextPOS}_{\text{finger}}$
- In reality
  - POSITION = (x, y)
  - Avoid icon surpassing finger



Optimize User Interaction Systematically

- What we need:
  - A well-established methodology
  - An engineering workload suite
  - An analysis/tuning toolkit
  - Sightings/requests/feedbacks from users

The key is to map user behavior into software metrics

Agenda

- Client device user interactions
- Android user interaction optimizations
  - Android Workload Suite and Android UXtune
- Case studies of Android optimizations
- Factors that impact Android user interactions
- Summary
- Information

Android Workload Suite (AWS)

- Goals
  - Reflect the representative usage of Android devices
  - Evaluate Performance, Power and User interactions
- AWS usages
  - Drive and validate Android optimizations
  - Support comparative and competitive analysis
- (Details in another slide deck)

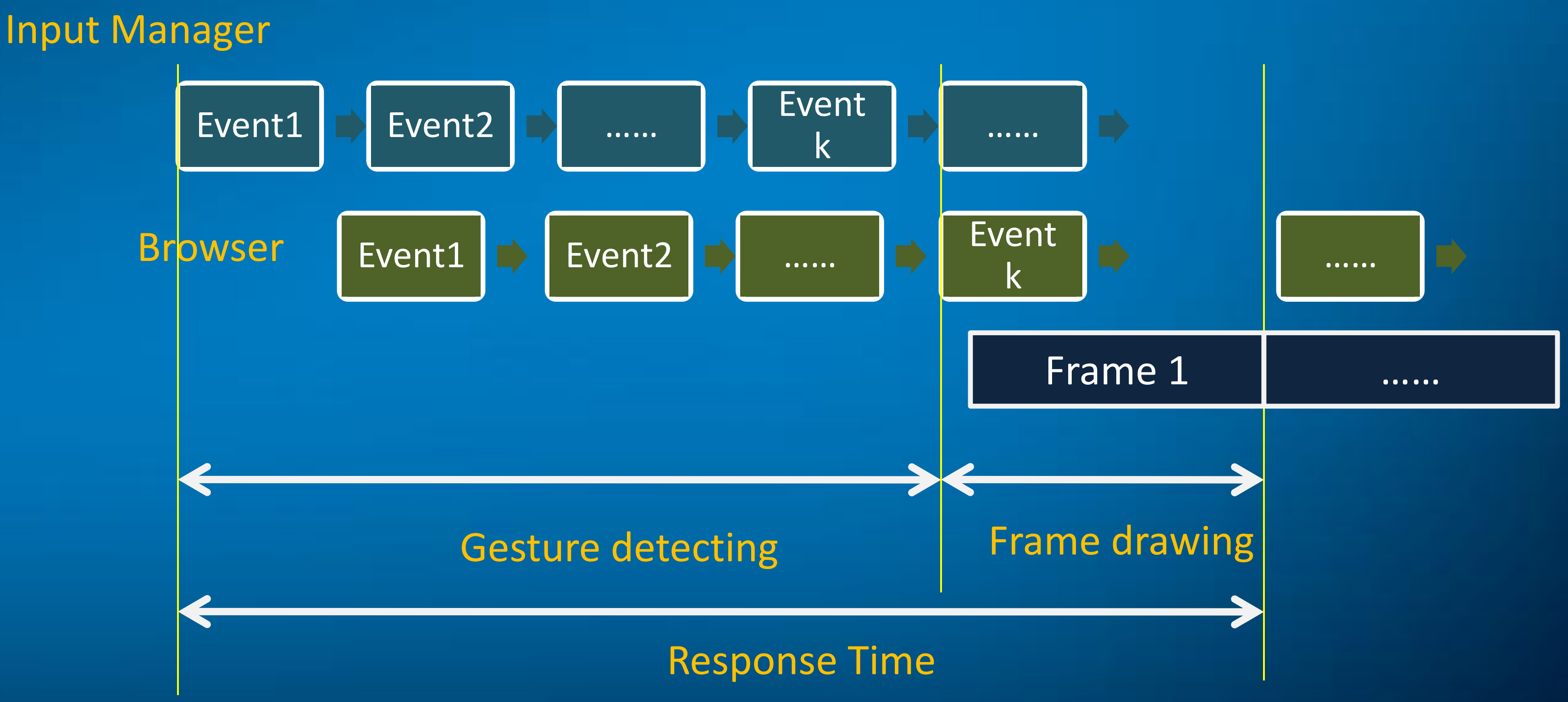
UXtune: An Analysis/Tuning Toolkit

- To analyze and optimize Android, we need
  - Repeatable inputs operating the device
    - Android input-Gestures
  - Sequence of interaction events between the system components, such as event, frame, thread, etc.
    - Android UXtune
  - Metrics outputs characterizing the behavior
    - Android meter-FPS
    - Android app-launch
    - Android touch-pressure
- (Details in another slide deck)

Agenda

- Business perspective and executive summary
- Client device user interactions
- Android user interaction optimizations
  - Android Workload Suite and Android UXtune
- Case studies of Android optimizations
- Factors that impact Android user interactions
- Summary
- Information

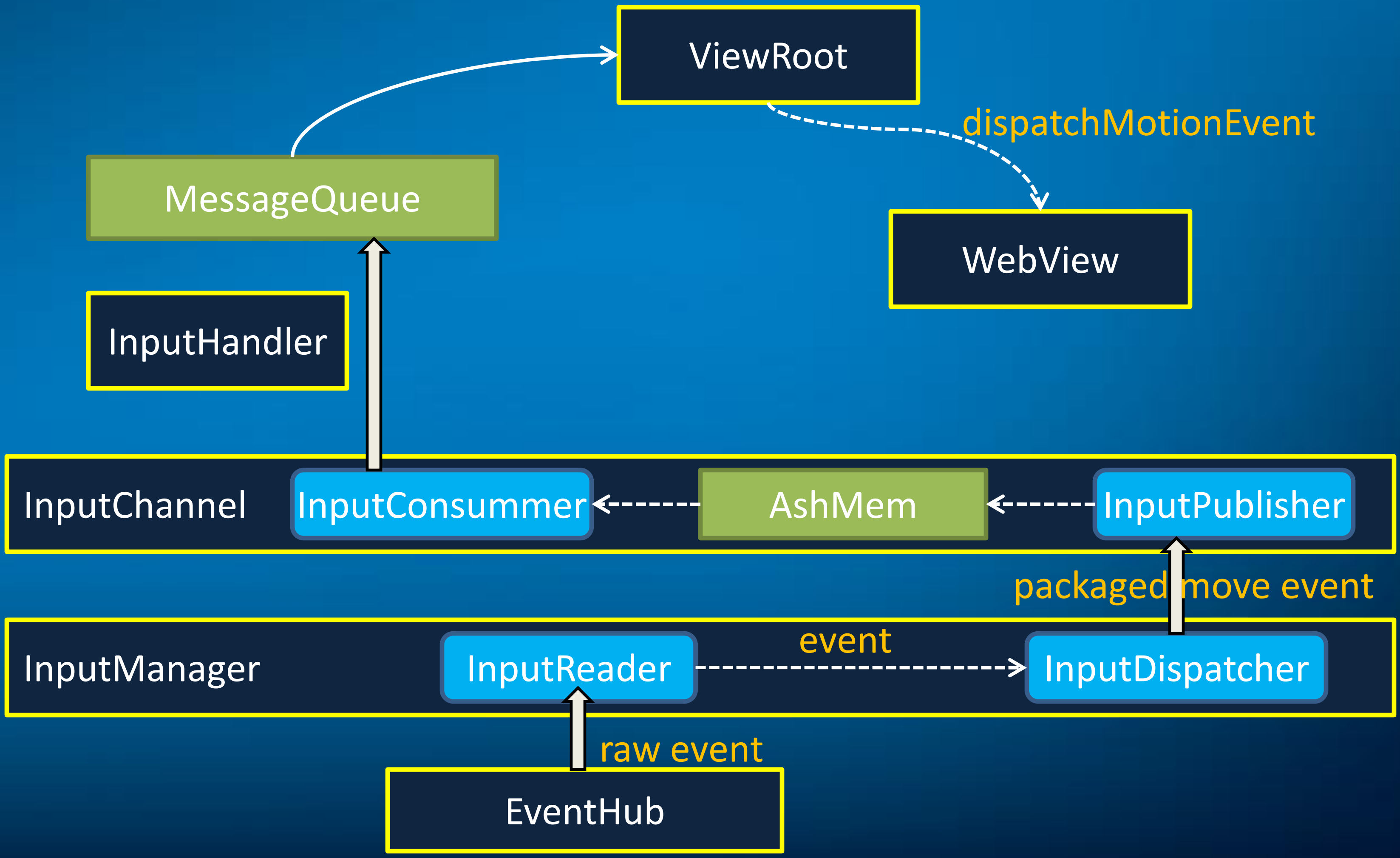
Browser Scroll Response Time



Scroll Response Time Optimizations

- Gesture detecting time
  - ACTION_DOWN event: record the start position
  - ACTION_MOVE event: compute the move Distance
  - Scroll is detected when Distance > Threadshold
- Frame drawing time
  - Skipped in this talk

User Input Event Dispatching



2011-09-12

Move Events Throttling

- InputDispatcher throttles the move-event emission rate
  - No need to emit move-event faster than platform maximal FPS
  - Set a minimum time interval between move-events dispatching , commonly 1/FPS
  - The move-events between two dispatches are grouped together

Experiment of Event Emission Throttling

- Without throttling, gesture detection time can be reduced by up to 1/fps in a common device

Time of Event	Throttling Delay	Time to Browser with throttling	Time to Browser Without throttling
0	0	2	2
6	12.096		8
15	3.096	21	17
25	12.14	41	27
45	11.894	59	46
71	4.191	79	72
78	15.355		79
87	6.355	98	88
96	**15.897**		**97**
106	5.897	**116**	

Events and gesture detection impact user interaction

Agenda

- Client device user interactions
- Android user interaction optimizations
  - Android Workload Suite and Android UXtune
- Case studies of Android optimizations
- Factors that impact Android user interactions
- Summary
- Information

Factors That Impact User Interactions

SOC factors

- Memory bus bandwidth, HW gfx acceleration
- Touch screen pressure resolution

OS/Runtime

- Multi-core software synchronization, load balance
- Thread scheduling priority, UI thread vs. other computations
- Runtime engine design

Input factors

- Event emission rate
- Sensor event detection
- Touch gesture detection

Animation design

- Property animation design
- Application transition
- Gesture/Sensor inputs response
- Operation smoothness
- Drag coherence

Agenda

- Client device user interactions
- Android user interaction optimizations
  - Android Workload Suite and Android UXtune
- Case studies of Android optimizations
- Factors that impact Android user interactions
- Summary
- Information

Summary

- Performance is not enough to characterize a client device
- An systematic approach has been established to engineering user interactions
  - AWS is a comprehensive engineering workload suite
  - UXtune is a toolkit to assist analysis and optimization
- Android user interactions can be impacted by the technical factors across the whole software stack

User Interaction is a new challenge but not insurmountable

Information

- Online resource for reference
  - http://ux.stackexchange.com/
  - http://www.useit.com/papers/responsetime.html
  - http://www.public.navy.mil/navsafecen/Documents/acquisition/MILSTD1472F.pdf
  - http://www.measuringux.com/