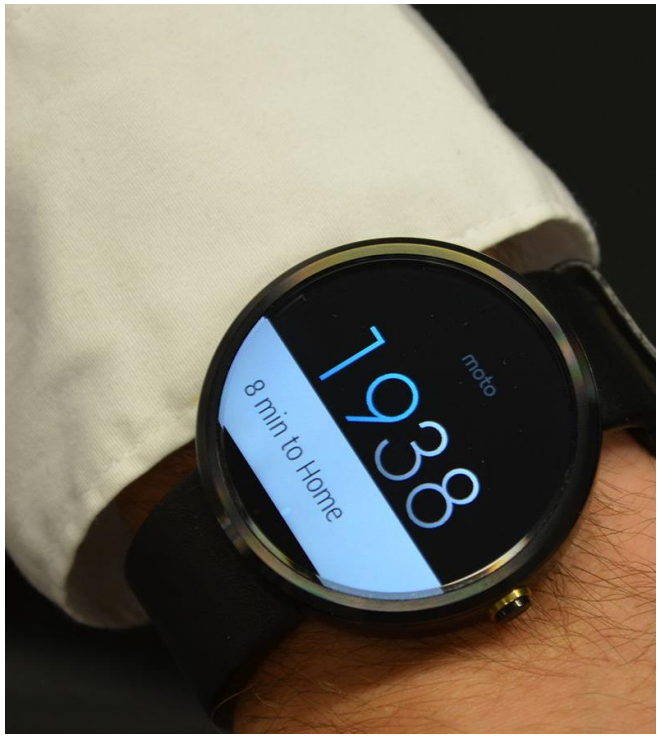
WearOS

ANDREA.FOSCHI13@STUDIO.UNIBO.IT

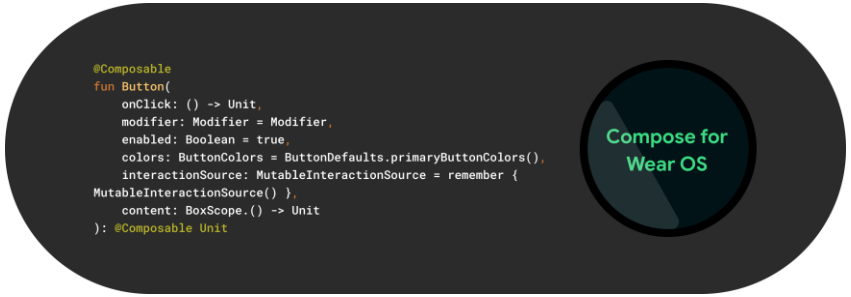
PROGRAMMAZIONE DI SISTEMI MOBILE

A.A. 2021/2022

Storia

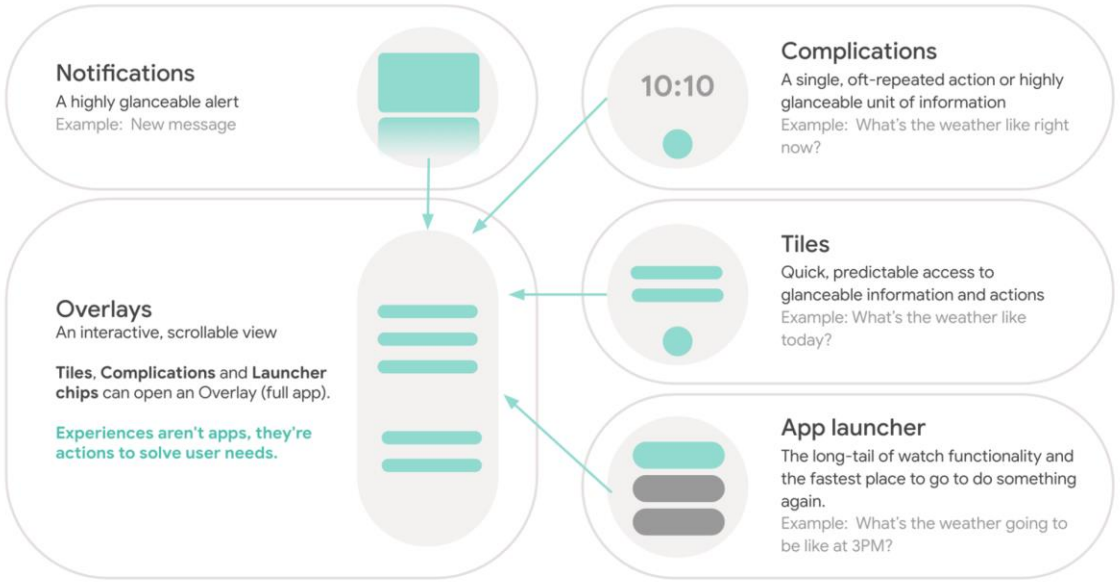


- **WearOS** è una versione di Android progettata per **smartwatch.**
- Compatibile anche per **iPhone** (iOS 8.2 o superiore).
- Nasce la prima versione il 25 giugno 2014 compatibile con 7 dispositivi.
- Richiede l'accoppiamento di uno smartphone Android 4.3 o superiore.
- L'ultima release è **WearOS 3** (11 agosto 2021).
- Permette la gestione e il controllo di diverse app:
  - **Routine** (Calendar, Pay, Keep, Maps);
  - **Contatti** (Messaggi, Gmail, Phone);
  - **Salute** (Fit, adidas Running, Sleep Cycle);
  - **Musica** (YouTube Music, Spoitfy, Pandora, iHeart Studio);
  - **Assistente Google.**



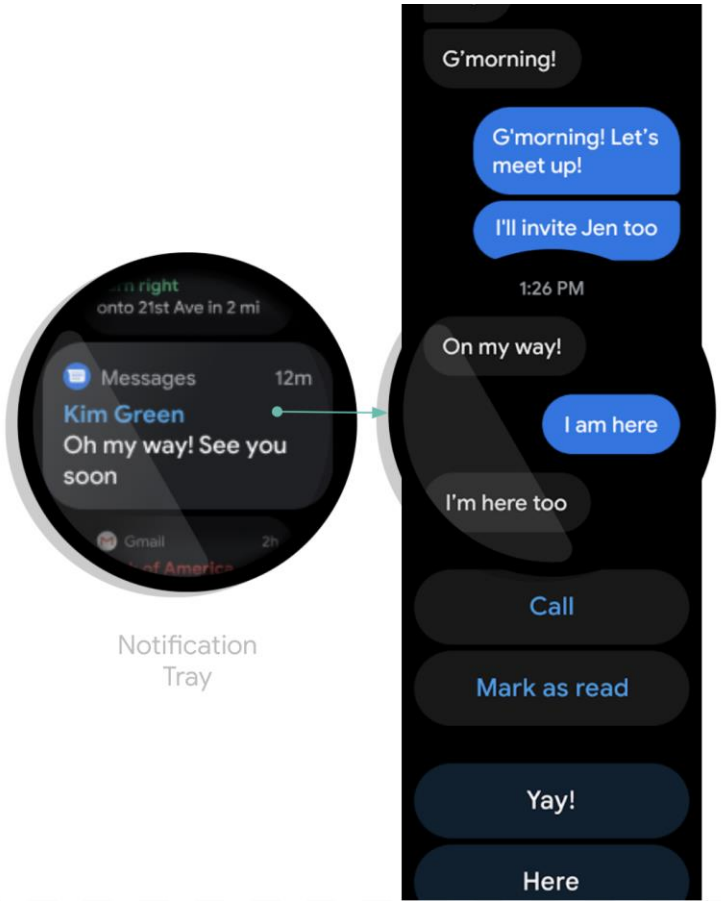
Compose for WearOS

- Fa parte di **Android Jetpack** e aiuta a scrivere codice migliore più velocemente;
- Compatibile su watch che usano WearOS 2.0 e 3.0;
- Per progettare le interfacce è consigliato l'uso di **WearComposeMaterial** siccome la versione mobile di Compose Material non è ottimizzata per i requisiti esclusivi di WearOS.
- Propone una serie di linee guide per lo sviluppo di tutte le possibili componenti (Navigation, Buttons, Cards, Dialogs, Lists, Pickers, Sliders…);



Principi di sviluppo per WearOS

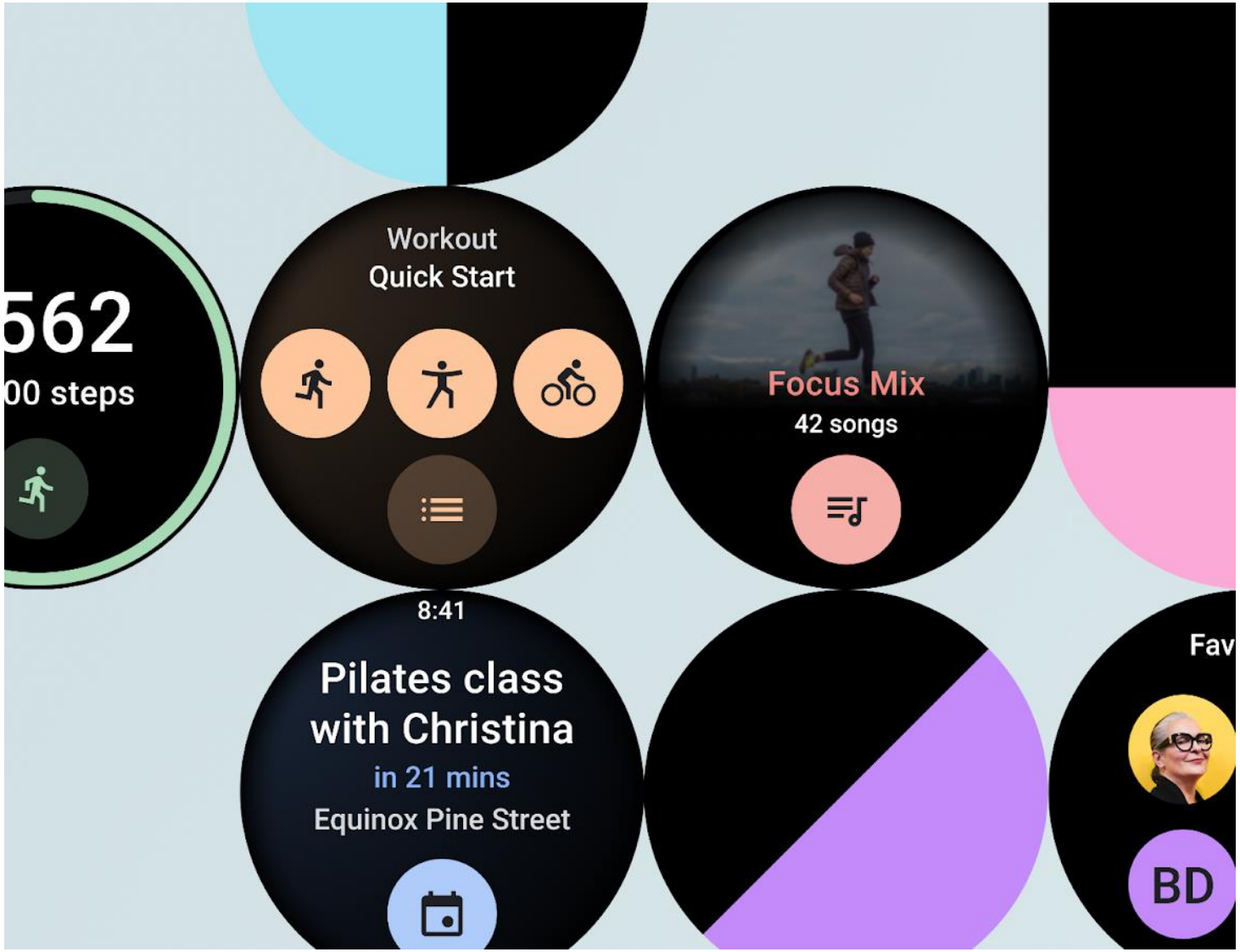
- Concentrarsi su una, massimo due **attività** anziché su un'esperienza completa dell'app;
- Aiutare l'utente a completare le attività sull'orologio in pochi secondi;
- Usare uno smartwatch deve aiutare gli utenti ad interagire con i propri dispositivi;
- Mantenere sul quadrante i contenuti pertinenti all'utente;
- Progettare applicazioni che funzionino anche **offline** e per connessioni lente;
- Nel caso l'attività non possa essere completata dal watch, aiutare l'utente a completarla da un altro dispositivo.



Notification Tray

Notifications

- Utilizzano le stesse API e hanno la stessa struttura delle notifiche sugli smartphones;
- Le notifiche vengono visualizzate sul watch perché vengono ricevute da un altro device e il sistema collega la notifica al watch, oppure perché il watch stesso ha creato la notifica;
- Per la creazione delle notifiche si usa la **NotificationCompat.Builder**;
- Si consiglia l'uso di notifiche espandibili come punto di partenza, in maniera da coinvolgere l'utente;
- Si possono aggiungere attività specifiche alle notifiche (come nascondere l'icona di un'app o di dettare un testo tramite l'input vocale) tramite il **WearableExtender**.



Tiles

- I **quadranti** sono l'unità fondamentale per questo tipo di applicazioni;
- Forniscono un facile accesso alle **informazioni** e alle **azioni** che gli utenti hanno bisogno per portare a termine le **operazioni**.
- L'utente sceglie quale quadrante visualizzare (timer, fintess, musica, messaggio…);
- Il Sistema Operativo gestisce il rendering dell'interfaccia utente del riquadro, nonché il layout e le risorse utilizzando il TileService.

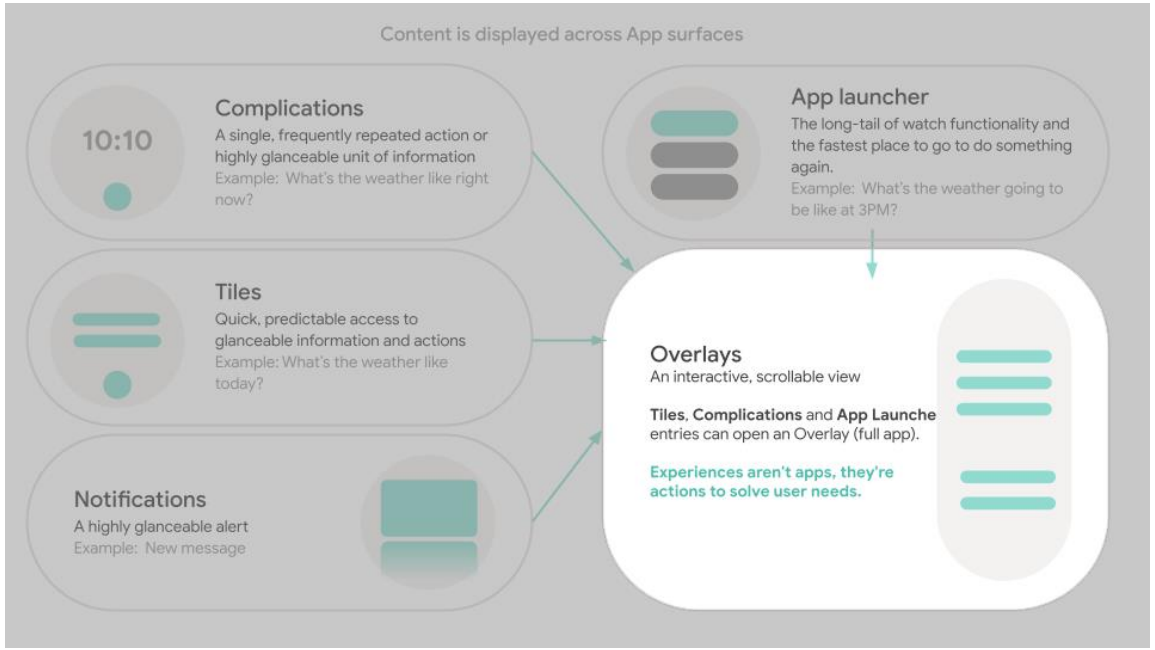
Complications

```
<service
    android:name=".provider.IncrementingNumberComplicationProviderService"
    android:icon="@drawable/icn_complications"
    android:label="@string/complications_provider_incrementing_number"
    android:permission="com.google.android.wearable.permission.BIND_COMPLICATION_PROVIDER">
    <intent-filter>
        <action
         android:name="android.support.wearable.complications.ACTION_COMPLICATION_UPDATE_REQUEST"/>
    </intent-filter>
</service>
```

- Le data provider apps espongono le informazioni alle **complications** dei quadranti, fornendo campi che contengono testo, stringhe, immagini, numeri;
- Un provider di dati si estende a un **ComplicationProviderService** per fornire dati utili direttamente su un quadrante;
- Per l'implementazione di un aggiornamento si usa un **BroadcastReceiver** che aggiorna i dati tramite il metodo **onComplicationUpdate()**;
- Bisogna dichiarare nel **manifest** le autorizzazioni per l'uso dei complications.

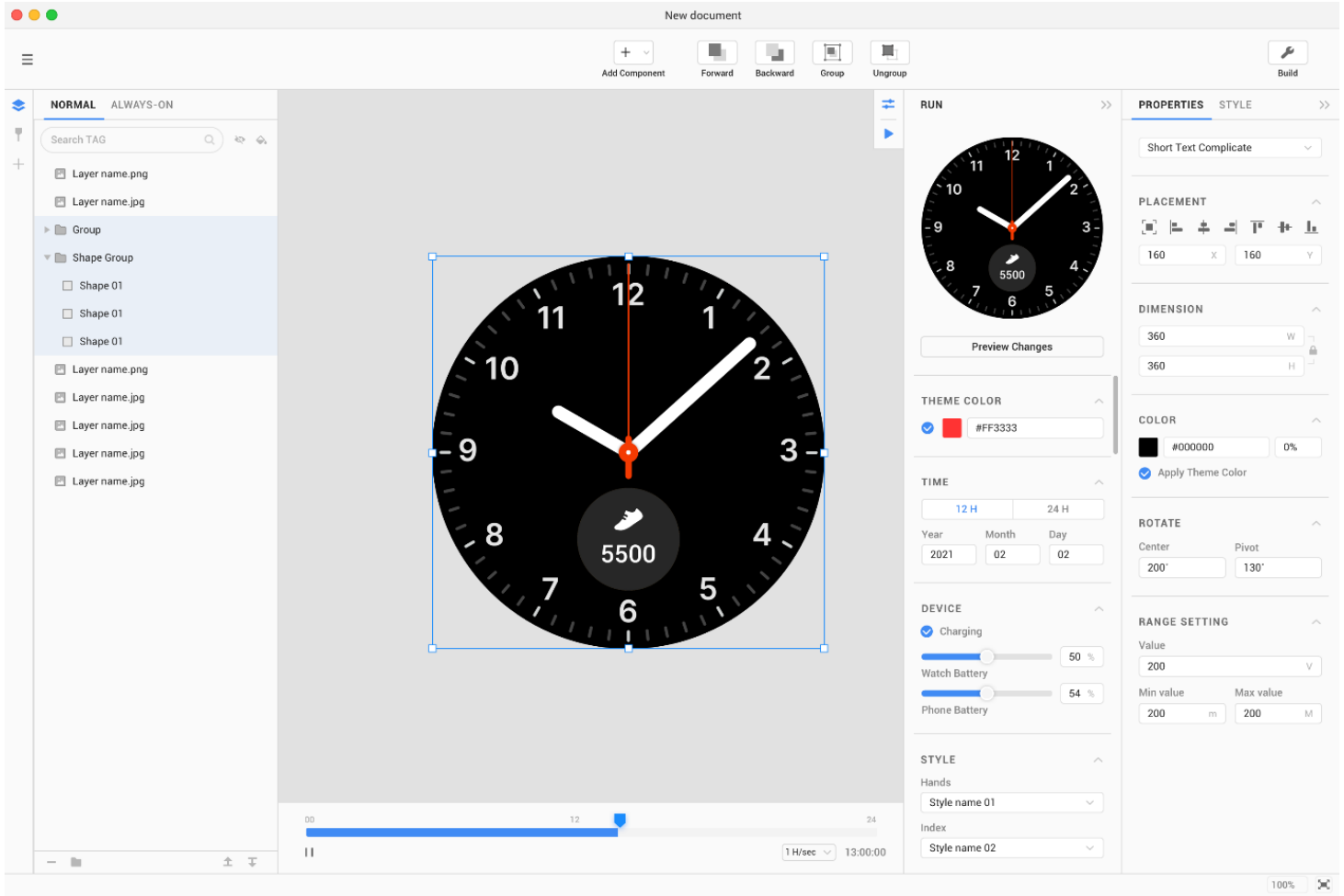
Input dell’utente

- **Pulsanti fisici:** un watch ha a sempre a disposizione un pulsante (quello di accensione), ma potrebbero essere presenti anche pulsanti multifunzione per svolgere delle azioni;
- **Rotary input:** alcuni dispositivi possiedono un rotore fisico che potrebbe permette, ad esempio, lo scorrimento;
- **Input Method Editor:** fornisce supporto a tastiere virtuali per inserire testo a mano o tramite gesti;
- **Voce:** ogni dispositivo è dotato di microfono, che permette di comunicare con la propria voce.



Overlays

- Un overlay è una **vista** focalizzata che dovrebbe gestire attività troppo complesse per una complication, tile o notifica;
- Le superfici che svolgono attività semplici si collegano a un **overlay** per consentire all'utente di svolgere un'attività complessa;
- Seguono tre principi per la UX:
  - **Focus:** si concentrano gli overlay sulle attività critiche per far svolgere attività in pochi secondi;
  - **Shallow and Linear:** evitare gerarchie superiori ai due livelli;
  - **Scroll:** favorire lo scorrimento per mostrare più contenuti sul watch.



Watch Face Studio

- Esiste un tool chiamato **Watch Face Studio** creato da Samsung che permette di creare quadranti relativi alle interfacce dell'applicazione senza la scrittura di codice;
- I quadranti creati con WFS possono essere scaricati ed installati per qualsiasi dispositivo WearOS 2.0 o superiore;
- Permette anche il testing immediato delle interfacce sia su device virtuali che fisici.