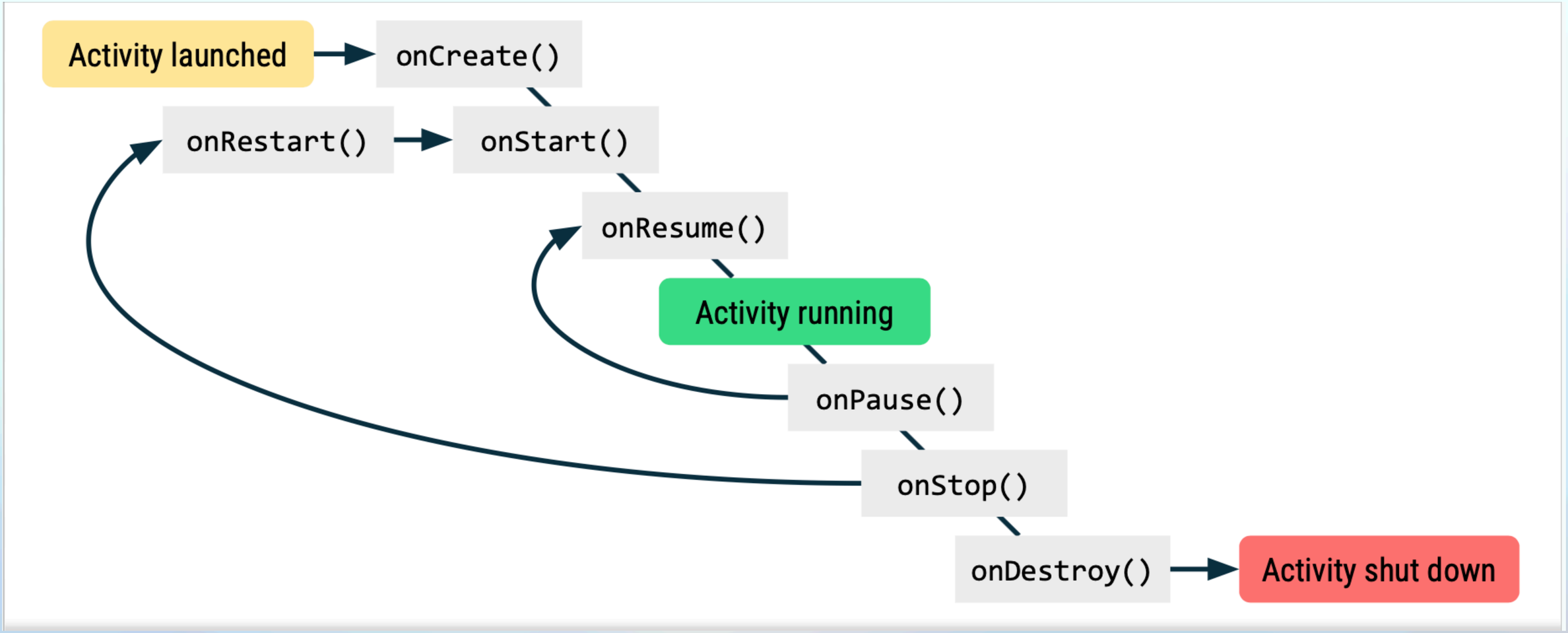
Fragment lifecycle & Navigation

Activity lifecycle - revisited



Summary of Activity States

State	Callbacks	Description
Created	`onCreate()`	Activity is being initialized.
Started	`onStart()`	Activity is visible to the user.
Resumed	`onResume()`	Activity has input focus.
Paused	`onPause()`	Activity does not have input focus.
Stopped	`onStop()`	Activity is no longer visible.
Destroyed	`onDestroy()`	Activity is destroyed.

Save state

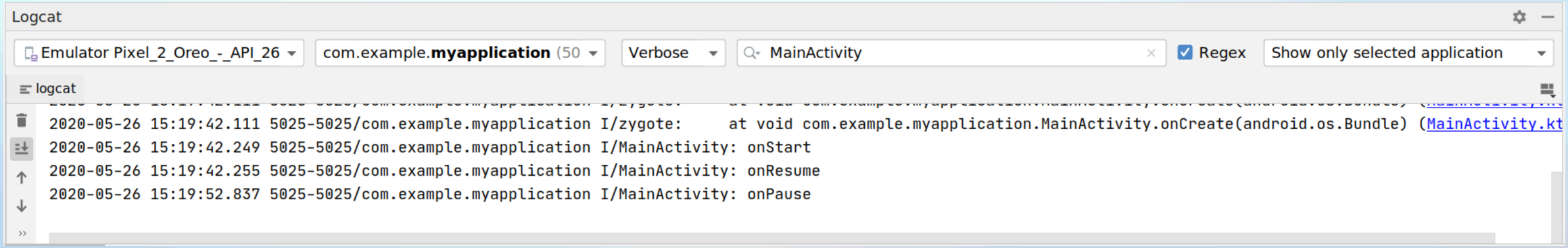
User expects UI state to stay the same after a config change or if the app is terminated when in the background.

- Activity is destroyed and restarted, or app is terminated and activity is started.
- Store user data needed to reconstruct app and activity Lifecycle changes:
  - **Use** Bundle **provided by** onSaveInstanceState()**.**
  - `onCreate()` receives the `Bundle` as an argument when activity is created again.

Logging

- Monitor the flow of events or state of your app.
- Use the built-in `Log` class or third-party library.
- **Example** Log **method call:** Log.d(TAG, “Message")

Syntax : Log.d(TAG, "Debug message”)



Write logs

Log Levels (in order of severity)

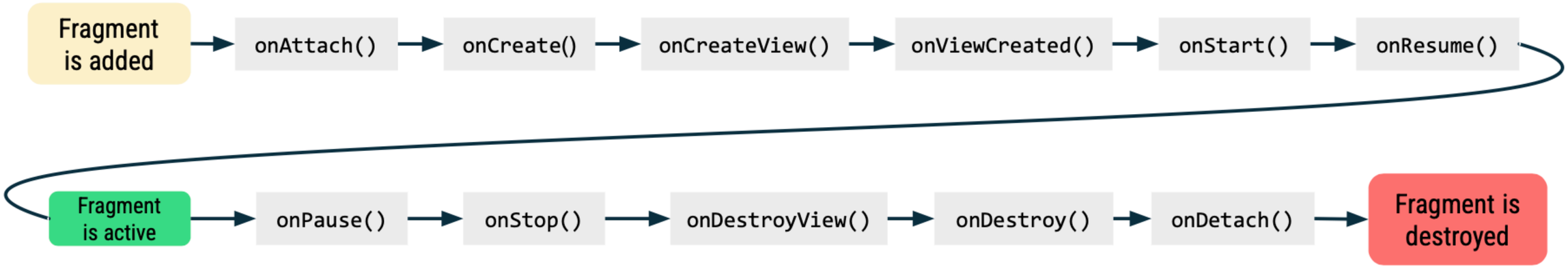
- **VERBOSE (V)**: Fine-grained details (lowest priority)
- **DEBUG (D)**: Debugging information
- **INFO (I)**: General runtime information
- **WARN (W)**: Potential issues that aren't immediate errors
- **ERROR (E)**: Errors and exceptions
- **ASSERT (A)**: Critical failures (highest priority)

Best practices

- Define a static TAG for each class (typically the class name)
- Log meaningful information with context
- Use appropriate log levels
- Remove or disable verbose logs in production
- Consider using a logging library for advanced needs
- Logging has performance overhead
- Use BuildConfig.DEBUG to conditionally log

Fragment lifecycle

Fragment States


Fragment is added
onAttach()
onCreate()
onCreateView()
onViewCreated()
onStart()
onResume()
Fragment is active
onPause()
onStop()
onDestroyView()
onDestroy()
onDetach()
Fragment is destroyed

Fragment Lifecycle methods

onAttach()

- Called when a fragment is attached to a context
- Immediately precedes `onCreate()`

onCreateView()

- Called to create the view hierarchy associated with the fragment
- Inflate the fragment layout here and return the root view

Fragment Lifecycle methods

onViewCreated()

- Called when view hierarchy has already been created
- Perform any remaining initialization here (for example, restore state from Bundle)

onDestroyView()

- is called when view hierarchy of fragment is removed.

onDetach()

- is called when fragment is no longer attached to the host.

Summary of Fragment Lifecycle

State	Callbacks	Description
Initialized	`onAttach()`	Fragment is attached to host.
Created	`onCreate()`, `onCreateView()`, `onViewCreated()`	Fragment is created and layout is being initialized.
Started	`onStart()`	Fragment is started and visible.
Resumed	`onResume()`	Fragment has input focus.
Paused	`onPause()`	Fragment no longer has input focus.
Stopped	`onStop()`	Fragment is not visible.
Destroyed	`onDestroyView()`, `onDestroy()`, `onDetach()`	Fragment is removed from host.

Thank you