Shoonya OMS – Order Lifecycle

```
ENTRY LIFECYCLE
----------------
TradingView Alert
 → process_alert()
 → ExecutionGuard.validate_and_prepare()
 → process_leg()
 → UniversalOrderCommand (ENTRY)
 → CommandService.submit()
 → ShoonyaBot.execute_command()
 → Broker.place_order()
 → OrderRecord CREATED
 → pending_commands (memory)


ADJUSTMENT LIFECYCLE
-------------------
Adjustment = EXIT (old leg) + ENTRY (new leg)

EXIT (Adjustment / Risk / SL):
request_exit()
 → UniversalOrderCommand (EXIT intent)
 → CommandService.register()
 → OrderWatcherEngine
 → _fire_exit()
 → LIMIT / MARKET auto-resolved
 → CommandService.submit()
 → Broker.place_order()

ENTRY (New leg):
Same as ENTRY lifecycle above


EXIT LIFECYCLE
--------------
Risk / SL / Trailing / Manual Trigger
 → request_exit()
 → EXIT intent registered
 → OrderWatcherEngine
 → LTP-based price discovery
 → requires_limit_order enforcement
 → Broker execution


BROKER RECONCILIATION
--------------------
OrderWatcherEngine
 → get_order_book()
 → COMPLETE / CANCEL / REJECT handling
 → DB + ExecutionGuard sync


CORE PRINCIPLES
---------------
• Intent ≠ Execution
• EXIT always watcher-driven
• ENTRY always alert-driven
• Adjustments are atomic EXIT + ENTRY
• Broker is the single source of truth
```