





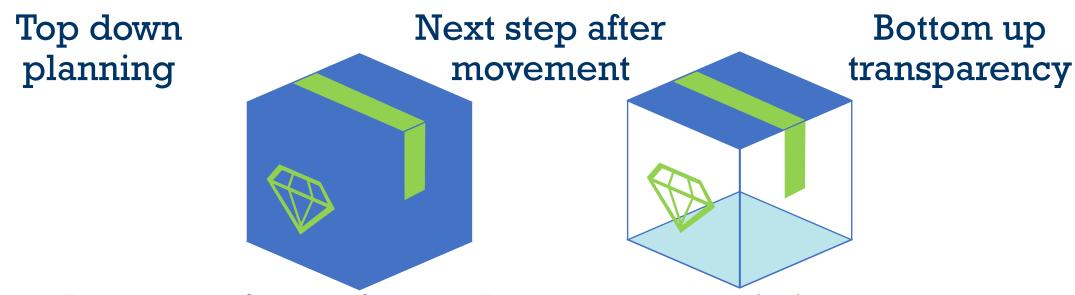




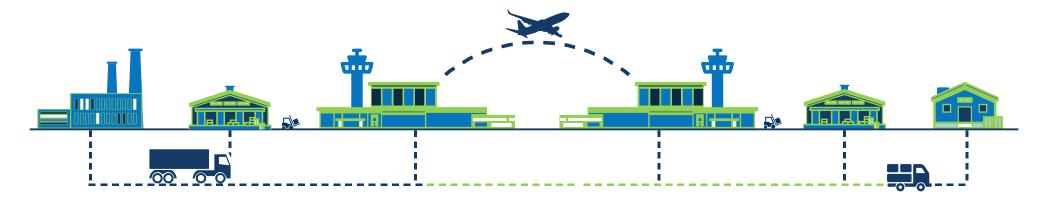




Care Mapping



Any type of care & complementary to existing programs





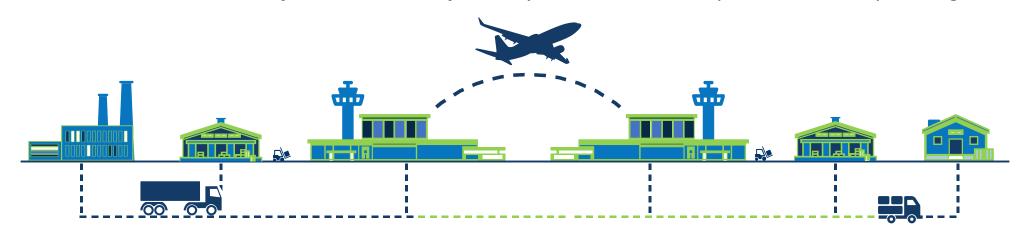






Planning & Control – Clear Care Events for every step

- Start with Check Procedures and Create Check Milestone for each operator (One level up)
- Top Down Planning for Shipment Custody and Operator Checks
- Bottom-Up planning for Care Actions to be applied, like checks movement in and out of zone, etc.
- Transparent and Planned Execution communicate results of Care Actions and replanning if needed
- CPC Care Protocol Check confirms the ability of the operator to handle the shipment. Before actual custody.
- CPA Care Protocol Applied recognizes custody by Operator and allows planning of care actions
- CPU Care Protocol Updates shows timing and outcome of care actions, allows re-planning if needed
- CPR Care Protocol Released reflects handover of custody and care to next operator, cancels planning.





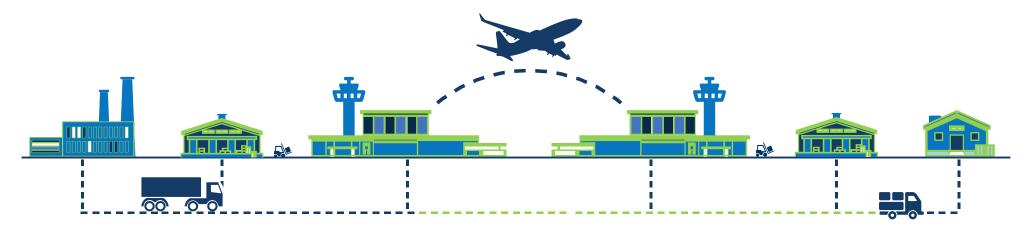






Identify Operators & Check Against Requirements

- 1 Create initial overview of operators are involved: warehouses, road and air carriers, GHA's, etc
- 2 Translate agreed requirements into verifiable abilities
 - Regulations What Care implications apply that need to be met?
 - Capabilities Can the chain provide, measure, treat and inform?
 - Capacity Can timings be met by process design, equipment and system capacity?
 - Compliance & Certification Need Relevant Certifications (CEIV, TAPA, GDP, Customer Specific, etc.)?
 - Allowed time total time, time allowed per operator and allowance for handovers
- 3 Execute Check and Verify Intended Operators, if possible through predefined product and service packs







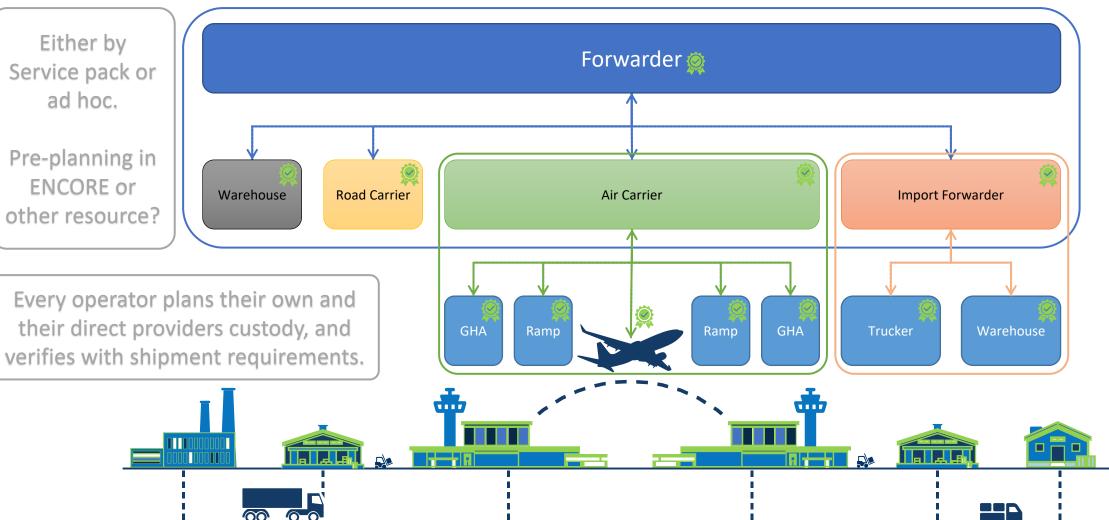




A Multilayer Approach to Requirements Check

Either by Service pack or ad hoc.

Pre-planning in **ENCORE** or other resource?



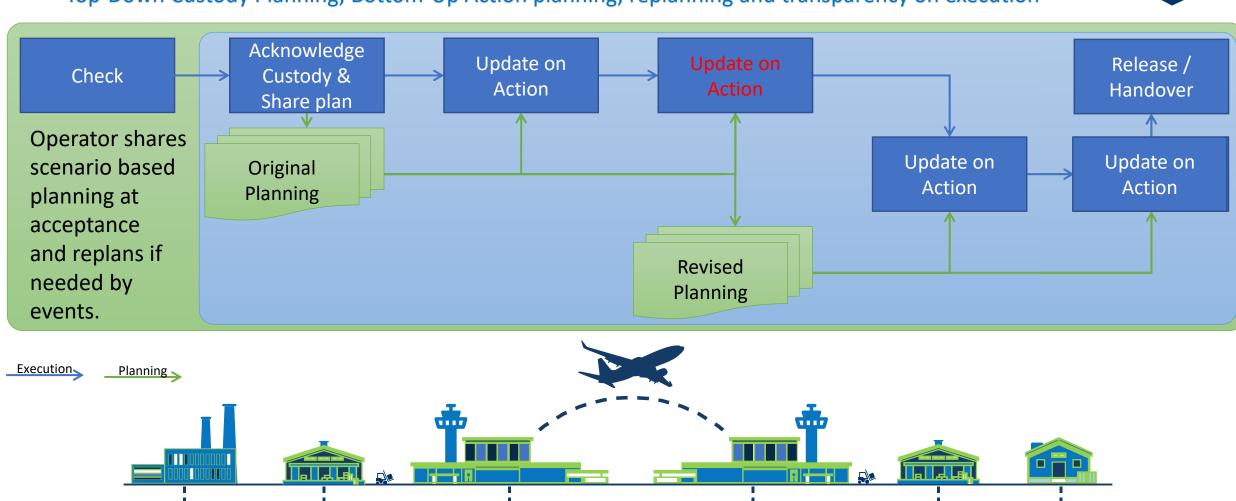








Top-Down Custody Planning; Bottom-Up Action planning, replanning and transparency on execution











Every custodian communicates upward on planning, replanning and execution

