Project Helios — Part 2

1) Project Requirements Document (PRD)

1.1 Scope Summary

- Mercury-anchored Dyson-swarm industrial base using seed robotic units that land, self-replicate via ISRU, and scale up to fabricate and launch reflector/collector panels into heliocentric orbits.
- Wireless power transmission: a small fraction is delivered to Earth (to meet Earth's needs); most harvested power is reinvested in space-infrastructure growth.
- Architecture scales toward near-full solar angular coverage over time.

1.2 Functional Requirements (FR)

Architecture & Autonomy

- **FR-A1** – Deploy seed multifunctional robots to Mercury and autonomously initialize ISRU and replication.
- **FR-A2** – Self-replicate structural/mechanical components locally; import complex electronics during Phases 1–2 (partial replication strategy).
- **FR-A3** – Fabricate, assemble, and launch reflector/collector modules into designated solar orbits without continuous ground control.
- **FR-A4** – Integrate wireless power transmission (laser/microwave) and Earth-side rectenna interfaces to deliver a small fraction of harvested energy to Earth.
- **FR-A5** – Provide decentralized, fault-tolerant control (payload-centric autonomy for assembly).

Resource & Manufacturing

- **FR-R1** – Map and extract Mercury metallic/silicate resources suitable for structural components and panel substrates.
- **FR-R2** – Operate electromagnetic mass drivers (or equivalent) for orbit insertion of panels.

Safety & Governance

- **FR-S1** – Replication must include hard limits, external kill-switch pathways, and environmental throttles.
- **FR-S2** – Comply with international frameworks (e.g., Outer Space Treaty/COSPAR) and project ethics.

1.3 Technical Requirements (TR)

Autonomy & Control

- **TR-AC1** – Payload-centric autonomy libraries for precision assembly (sub-mm fit-up at assembly cells).
- **TR-AC2** – Mesh networking with degradation-tolerant protocols and local fallback modes.

ISRU & Manufacturing

- **TR-IM1** – ISRU lines for metal extraction (carbothermal/electrolysis) and additive manufacturing of frames/panels.
- **TR-IM2** – Foundry cells hardened for Mercury thermal/radiation extremes and dust.

Launch & Orbital

- **TR-LO1** – Mass drivers accelerate panel stacks to ≥4.3 km/s (Mercury escape ~4.25 km/s) for heliocentric transfer.
- **TR-LO2** – Panels include station-keeping and attitude control for formation maintenance.

Power & Beaming

- **TR-PB1** – Collector modules use high-efficiency PV or reflector+thermal PV sized for inner-solar flux.
- **TR-PB2** – Space→Earth beaming chain achieves net ≥40% transfer efficiency at rectenna boundary by Phase-4 pilots.

Safety & Governance

- **TR-SG1** – Replication caps, resource quotas, mission-lifetime deactivation for units/factories.
- **TR-SG2** – Audit logs of autonomous actions; update controls with formal-verification gates.

1.4 Performance Requirements (PR)

- **PR-1** – By 2028-03: closed-loop replication (structural components) ≥1 unit/week (analog); by 2031-09 in-situ replication ≥1 unit/day.
- **PR-2** – Per factory node: ≥10 t/day processed regolith by 2031; scalable to ≥100 t/day by 2037.
- **PR-3** – ≥10 MW(e)/day panel output per mature fabrication cluster by 2037.
- **PR-4** – Mass-driver cadence ≥50 panel-stack launches/day by Phase 3.
- **PR-5** – ≥20 TW net delivered to Earth by 2045-09; remainder used for space infrastructure.
- **PR-6** – Validate formation mechanics and manufacturing scale-up toward near-full solar coverage.
- **PR-7** – Tolerate ≥15% unit attrition/year without net capacity loss via redundancy + replication.

1.5 V&V Strategy

- **V-1** Ground demos (assembly accuracy, ISRU batches, replication dry-run).
- **V-2** Mercury pathfinder: mining→parts→assembly→limited replication.
- **V-3** Panel pilot + mass-driver demo to heliocentric orbit.
- **V-4** Power-beaming pilot to Earth rectenna.

2) Work Breakdown Structure (WBS)

The WBS below structures scope into major deliverable areas and work packages used for planning and scheduling. The visuals provide multiple ways to understand scope distribution.

In [3]:

```python
# %%
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# Define comprehensive WBS hierarchy: Project -> Phase/Deliverable -> Work Package -> Activities
wbs_data = [
    # Root Project Level
    ("Project Helios", "", "", "", 100, "Project"),

    # Phase/Deliverable Level (Level 1)
    ("Phase 1: Seed Deployment", "Project Helios", "", "", 25, "Phase"),
    ("Phase 2: Bootstrap Replication", "Project Helios", "", "", 25, "Phase"),
    ("Phase 3: Infrastructure Expansion", "Project Helios", "", "", 25, "Phase"),
    ("Phase 4: Dyson Swarm Assembly", "Project Helios", "", "", 15, "Phase"),
    ("Phase 5: Operations", "Project Helios", "", "", 10, "Phase"),

    # Work Package Level (Level 2) - Phase 1
    ("1.0 Program Mgmt & SE", "Phase 1: Seed Deployment", "", "", 8, "Work Package"),
    ("2.0 Seed Units R&D", "Phase 1: Seed Deployment", "", "", 12, "Work Package"),
    ("8.0 Verification & Pilots", "Phase 1: Seed Deployment", "", "", 5, "Work Package"),

    # Work Package Level (Level 2) - Phase 2
    ("3.0 ISRU & Manufacturing", "Phase 2: Bootstrap Replication", "", "", 15, "Work Package"),
    ("4.0 Replication & Assembly Ops", "Phase 2: Bootstrap Replication", "", "", 10, "Work Package"),

    # Work Package Level (Level 2) - Phase 3
    ("5.0 Launch & Orbital Integration", "Phase 3: Infrastructure Expansion", "", "", 8, "Work Package"),
    ("6.0 Collector/Reflector Panels", "Phase 3: Infrastructure Expansion", "", "", 10, "Work Package"),
    ("7.0 Power Beaming & Earth Interface", "Phase 3: Infrastructure Expansion", "", "", 7, "Work Package"),

    # Work Package Level (Level 2) - Phase 4
    ("9.0 Deployment & Scale-Up", "Phase 4: Dyson Swarm Assembly", "", "", 15, "Work Package"),

    # Activity Level (Level 3) - detailed breakdown
    # 1.0 Program Mgmt & SE Activities
    ("1.1 Governance/Stakeholders", "Phase 1: Seed Deployment", "1.0 Program Mgmt & SE", "", 3, "Activity"),
    ("1.2 Systems Eng & CM", "Phase 1: Seed Deployment", "1.0 Program Mgmt & SE", "", 3, "Activity"),
    ("1.3 Safety & Replication Governance", "Phase 1: Seed Deployment", "1.0 Program Mgmt & SE", "", 2, "Activity"),

    # 2.0 Seed Units R&D Activities
    ("2.1 Mechatronics & Power", "Phase 1: Seed Deployment", "2.0 Seed Units R&D", "", 4, "Activity"),
    ("2.2 Autonomy Stack", "Phase 1: Seed Deployment", "2.0 Seed Units R&D", "", 4, "Activity"),
    ("2.3 Partial Replication Design", "Phase 1: Seed Deployment", "2.0 Seed Units R&D", "", 2, "Activity"),
    ("2.4 Environmental Testing", "Phase 1: Seed Deployment", "2.0 Seed Units R&D", "", 2, "Activity"),

    # 3.0 ISRU & Manufacturing Activities
    ("3.1 Resource Mapping", "Phase 2: Bootstrap Replication", "3.0 ISRU & Manufacturing", "", 3, "Activity"),
    ("3.2 Mining/Extraction", "Phase 2: Bootstrap Replication", "3.0 ISRU & Manufacturing", "", 5, "Activity"),
    ("3.3 Processing Lines", "Phase 2: Bootstrap Replication", "3.0 ISRU & Manufacturing", "", 4, "Activity"),
    ("3.4 Additive Mfg Cells", "Phase 2: Bootstrap Replication", "3.0 ISRU & Manufacturing", "", 2, "Activity"),
    ("3.5 QA & Metrology", "Phase 2: Bootstrap Replication", "3.0 ISRU & Manufacturing", "", 1, "Activity"),

    # 4.0 Replication & Assembly Ops Activities
    ("4.1 Assembly Cells", "Phase 2: Bootstrap Replication", "4.0 Replication & Assembly Ops", "", 4, "Activity"),
    ("4.2 Replication Controls", "Phase 2: Bootstrap Replication", "4.0 Replication & Assembly Ops", "", 3, "Activity"),
    ("4.3 Spares & Maintenance", "Phase 2: Bootstrap Replication", "4.0 Replication & Assembly Ops", "", 3, "Activity"),

    # 5.0 Launch & Orbital Integration Activities
    ("5.1 Mass Driver", "Phase 3: Infrastructure Expansion", "5.0 Launch & Orbital Integration", "", 4, "Activity"),
    ("5.2 Panel Packaging & Launch", "Phase 3: Infrastructure Expansion", "5.0 Launch & Orbital Integration", "", 2, "Activity"),
    ("5.3 Orbital Insertion & Formation", "Phase 3: Infrastructure Expansion", "5.0 Launch & Orbital Integration", "", 2, "Activity"),

    # 6.0 Collector/Reflector Panels Activities
    ("6.1 Bus/Thermal/Coatings", "Phase 3: Infrastructure Expansion", "6.0 Collector/Reflector Panels", "", 3, "Activity"),
    ("6.2 PV/Reflector Integration", "Phase 3: Infrastructure Expansion", "6.0 Collector/Reflector Panels", "", 4, "Activity"),
    ("6.3 Health/Telemetry", "Phase 3: Infrastructure Expansion", "6.0 Collector/Reflector Panels", "", 3, "Activity"),

    # 7.0 Power Beaming & Earth Interface Activities
    ("7.1 Space Transmitters", "Phase 3: Infrastructure Expansion", "7.0 Power Beaming & Earth Interface", "", 3, "Activity"),
    ("7.2 Rectenna & Grid", "Phase 3: Infrastructure Expansion", "7.0 Power Beaming & Earth Interface", "", 2, "Activity"),
    ("7.3 Regulatory & Safety", "Phase 3: Infrastructure Expansion", "7.0 Power Beaming & Earth Interface", "", 2, "Activity"),

    # 8.0 Verification & Pilots Activities
    ("8.1 Ground Demos", "Phase 1: Seed Deployment", "8.0 Verification & Pilots", "", 2, "Activity"),
    ("8.2 Mercury Pathfinder", "Phase 1: Seed Deployment", "8.0 Verification & Pilots", "", 2, "Activity"),
    ("8.3 Panel & Beaming Pilots", "Phase 1: Seed Deployment", "8.0 Verification & Pilots", "", 1, "Activity"),

    # 9.0 Deployment & Scale-Up Activities
    ("9.1 Fab Ramp", "Phase 4: Dyson Swarm Assembly", "9.0 Deployment & Scale-Up", "", 5, "Activity"),
    ("9.2 Multi-Driver Network", "Phase 4: Dyson Swarm Assembly", "9.0 Deployment & Scale-Up", "", 5, "Activity"),
    ("9.3 Ops/Maint/Optimization", "Phase 4: Dyson Swarm Assembly", "9.0 Deployment & Scale-Up", "", 5, "Activity"),
]

wbs_df = pd.DataFrame(wbs_data, columns=["Element", "Phase", "Work_Package", "Parent", "Effort", "Level"])

# Create hierarchical tree structure for visualization
def create_tree_paths(df):
    """Create proper hierarchical paths for treemap visualization"""
    tree_data = []

    for _, row in df.iterrows():
        if row['Level'] == 'Project':
            tree_data.append({
                'ids': row['Element'],
                'labels': row['Element'],
                'parents': '',
                'values': row['Effort'],
                'level': 'Project'
            })
        elif row['Level'] == 'Phase':
            tree_data.append({
                'ids': row['Element'],
                'labels': row['Element'],
                'parents': 'Project Helios',
                'values': row['Effort'],
                'level': 'Phase'
            })
        elif row['Level'] == 'Work Package':
            tree_data.append({
                'ids': row['Element'],
                'labels': row['Element'],
                'parents': row['Phase'],
                'values': row['Effort'],
                'level': 'Work Package'
            })
        elif row['Level'] == 'Activity':
            tree_data.append({
                'ids': row['Element'],
                'labels': row['Element'],
                'parents': row['Work_Package'],
                'values': row['Effort'],
                'level': 'Activity'
            })

    return pd.DataFrame(tree_data)

tree_df = create_tree_paths(wbs_df)

# Create Hierarchical Treemap
fig_tree = go.Figure(go.Treemap(
    ids=tree_df['ids'],
    labels=tree_df['labels'],
    parents=tree_df['parents'],
    values=tree_df['values'],
    branchvalues="total",
    hovertemplate='<b>%{label}</b><br>Level: %{customdata}<br>Effort: %{value}%<extra></extra>',
    customdata=tree_df['level'],
    maxdepth=4,
    textinfo="label+value",
))

fig_tree.update_layout(
    title="WBS Hierarchical Tree Structure<br><sub>Project → Phases → Work Packages → Activities</sub>",
    font_size=10,
    margin=dict(t=80, l=0, r=0, b=0),
    height=600
)

fig_tree.show()

# Create Sunburst Chart for alternative tree visualization
fig_sun = go.Figure(go.Sunburst(
    ids=tree_df['ids'],
    labels=tree_df['labels'],
    parents=tree_df['parents'],
    values=tree_df['values'],
    branchvalues="total",
    hovertemplate='<b>%{label}</b><br>Level: %{customdata}<br>Effort: %{value}%<extra></extra>',
    customdata=tree_df['level'],
    maxdepth=4,
))

fig_sun.update_layout(
    title="WBS Sunburst View<br><sub>Radial Tree Structure</sub>",
    font_size=12,
    margin=dict(t=80, l=0, r=0, b=0),
    height=600
)

fig_sun.show()
```

3) Activity List (Selected Work Packages)

WP ID	Activity	Duration	Required Skills
1.1	Program kickoff, governance framework & ethics controls	8 weeks	Program Mgmt, Space Law, Ethics
2.1	Seed robot mechatronics breadboard (Gen-0)	20 weeks	Mech/EE, Thermal, Dust Mitigation
2.2	Autonomy stack v1 (payload-centric, local behaviors)	24 weeks	Robotics/Controls, CV, SW Eng
2.4	Environmental testing (TVAC/rad/abrasion)	16 weeks	Test Eng, Materials, QA
3.1	Mercury site selection & resource mapping plan	12 weeks	Planetary Science, Geology, GNC
3.2	Mining subsystem prototype (excavation + haul)	28 weeks	Mining Eng, Mechatronics
3.3	Metal extraction line (carbothermal/electrolysis)	36 weeks	ChemE, Materials, Process Eng
3.4	Additive manufacturing cell (metal AM)	28 weeks	AM/Metallurgy, NDE/Metrology
4.1	Assembly cell design (fixtures, jigs, toolchains)	18 weeks	Mfg Eng, Robotics
4.2	Replication controls (caps, quotas, kill-switch)	12 weeks	Safety Eng, Formal Methods
5.1	Mass driver conceptual → PDR	24 weeks	Power, EM, Structures
5.2	Mass driver prototype coil & firing test	20 weeks	EM, HV Power, Controls
6.1	Panel bus + thermal design, breadboard	20 weeks	Thermal, Power, Mech
6.2	PV/reflector selection & coatings validation	24 weeks	PV/Optics, Materials
7.1	Beaming chain architecture (space + Earth)	14 weeks	RF/Photonics, Power Systems
7.2	Rectenna pilot site studies & permits	16 weeks	Grid Interconnect, Policy
8.1	Ground integrated demo (replication dry-run)	16 weeks	IV&V, Systems
8.2	Mercury pathfinder integration & launch prep	28 weeks	AIT/ATLO, Mission Ops
9.1	Panel fab line ramp plan & ops playbook	18 weeks	Industrial Eng, Ops
9.2	Ops, maintenance, and optimization loop (ongoing)	Continuous	Ops, Data/ML

4) Project Milestones

#	Milestone	Target Date	Owner
M1	Part-2 Baseline Approved (this package)	2025-09-28	PMO
M2	Seed Gen-0 prototype functional	2026-05-15	Seed R&D Lead
M3	Ground demo: partial replication (structural)	2026-11-30	IV&V Lead
M4	Mass driver PDR complete	2027-03-31	Launch Systems Lead
M5	Payload-centric autonomy v1 validated	2027-06-30	Autonomy Lead
M6	Mercury pathfinder launch	2028-09-30	Mission Director
M7	In-situ replication ≥1 unit/day	2031-09-30	Surface Ops Lead
M8	First reflector panels launched to solar orbit	2034-06-30	Launch Ops
M9	Power-beaming pilot to Earth rectenna	2036-12-31	Beaming Lead
M10	Industrial scale-up (≥10 MW(e)/day per cluster)	2037-09-30	Mfg Director
M11	Multi-driver launch network online	2040-03-31	Launch Systems
M12	Phase-4 energy delivery to Earth (sustained)	2043-09-30	Grid Liaison
M13	≥20 TW net delivered to Earth	2045-09-30	Program Exec

In [5]:

```
# %%
# Milestone lollipop timeline
import pandas as pd
import plotly.graph_objects as go

milestones = [
    ("M1","2025-09-28","PMO"),
    ("M2","2026-05-15","Seed R&D Lead"),
    ("M3","2026-11-30","IV&V Lead"),
    ("M4","2027-03-31","Launch Systems Lead"),
    ("M5","2027-06-30","Autonomy Lead"),
    ("M6","2028-09-30","Mission Director"),
    ("M7","2031-09-30","Surface Ops Lead"),
    ("M8","2034-06-30","Launch Ops"),
    ("M9","2036-12-31","Beaming Lead"),
    ("M10","2037-09-30","Mfg Director"),
    ("M11","2040-03-31","Launch Systems"),
    ("M12","2043-09-30","Grid Liaison"),
    ("M13","2045-09-30","Program Exec"),
]

ms_df = pd.DataFrame(milestones, columns=["Milestone","Date","Owner"])
ms_df['Date'] = pd.to_datetime(ms_df['Date'])
ms_df['y'] = range(1, len(ms_df)+1)

fig = go.Figure()
fig.add_trace(go.Scatter(x=ms_df['Date'], y=ms_df['y'], mode='markers',
                         marker=dict(size=12, color='#1f77b4'),
                         text=ms_df['Milestone'] + " — " + ms_df['Owner'],
                         hovertemplate="%{text}<br>%{x|%Y-%m-%d}<extra></extra>"))
for i, row in ms_df.iterrows():
    fig.add_shape(type='line', x0=row['Date'], x1=row['Date'], y0=0, y1=row['y'], line=dict(color='#1f77b4', width=2))

fig.update_layout(title='Project Milestones — Timeline (Lollipop)', xaxis_title='Date', yaxis=dict(title='', showticklabels=False), height=480)
fig.show()
```

5) Activity-Based Schedule (by Phases)

5.1 Phase Overview

Phase	Phase Name	Start Date	End Date	Duration
1	Seed Deployment	2025-10-01	2028-09-30	3.0 years
2	Bootstrap Replication	2028-10-01	2031-09-30	3.0 years
3	Infrastructure Expansion	2031-10-01	2037-09-30	6.0 years
4	Dyson Swarm Assembly	2037-10-01	2043-09-30	6.0 years
5	Full Operational Capability	2043-10-01	2045-09-30	2.0 years

5.2 Detailed Activity Schedule

| Activity ID | Activity Name | Phase | Start Date | Finish Date | Duration (weeks) | Work Package | |---|---|---|---|---|---|---| | **Phase 1 Activities** | | 1.1.1 | Program setup & ethics governance | Phase 1 | 2025-10-01 | 2025-11-26 | 8 | Program Mgmt & SE | | 2.1.1 | Seed Gen-0 mechatronics | Phase 1 | 2025-11-27 | 2026-04-14 | 20 | Seed Units R&D | | 2.2.1 | Autonomy v1 (payload-centric) | Phase 1 | 2026-01-15 | 2026-06-30 | 24 | Seed Units R&D | | 2.4.1 | Environmental testing (seed units) | Phase 1 | 2026-07-01 | 2026-10-20 | 16 | Seed Units R&D | | 8.1.1 | Ground replication dry-run | Phase 1 | 2026-10-21 | 2027-02-08 | 16 | Verification & Pilots | | 5.1.1 | Mass driver PDR | Phase 1 | 2026-10-21 | 2027-04-04 | 24 | Launch & Orbital Integration | | 8.2.1 | Pathfinder AIT/launch prep | Phase 1 | 2028-01-01 | 2028-07-15 | 28 | Verification & Pilots | | **Phase 2 Activities** | | 3.1.1 | Site selection & resource mapping | Phase 2 | 2028-10-01 | 2029-03-31 | 26 | ISRU & Manufacturing | | 3.2.1 | Mining + processing line v1 | Phase 2 | 2029-04-01 | 2030-03-31 | 52 | ISRU & Manufacturing | | 3.4.1 | Additive mfg cell v1 | Phase 2 | 2029-07-01 | 2030-03-31 | 39 | ISRU & Manufacturing | | 4.1.1 | Assembly cell design & build | Phase 2 | 2030-04-01 | 2030-12-31 | 39 | Replication & Assembly | | 4.2.1 | Replication controls implementation | Phase 2 | 2030-10-01 | 2031-06-30 | 39 | Replication & Assembly | | **Phase 3 Activities** | | 6.1.1 | Panel fab line (cluster 1) | Phase 3 | 2031-10-01 | 2032-07-07 | 40 | Collector/Reflector Panels | | 5.1.2 | Mass driver prototype build & fire | Phase 3 | 2032-01-01 | 2032-10-08 | 40 | Launch & Orbital Integration | | 6.2.1 | PV/reflector integration testing | Phase 3 | 2032-07-08 | 2033-06-30 | 51 | Collector/Reflector Panels | | 7.1.1 | Space transmitter development | Phase 3 | 2033-01-01 | 2034-06-30 | 78 | Power Beaming & Earth Interface | | 5.2.1 | Panel packaging & launch system | Phase 3 | 2034-01-01 | 2035-06-30 | 78 | Launch & Orbital Integration | | 7.2.1 | Rectenna pilot site development | Phase 3 | 2035-01-01 | 2036-06-30 | 78 | Power Beaming & Earth Interface | | 8.3.1 | Power-beaming pilot | Phase 3 | 2036-06-01 | 2036-11-30 | 26 | Verification & Pilots | | **Phase 4 Activities** | | 9.1.1 | Multi-cluster fab ramp | Phase 4 | 2037-10-01 | 2040-10-01 | 156 | Deployment & Scale-Up | | 9.2.1 | Multi-driver network online | Phase 4 | 2039-07-01 | 2041-01-15 | 80 | Deployment & Scale-Up | | 9.1.2 | Industrial scale-up operations | Phase 4 | 2041-01-16 | 2043-06-30 | 128 | Deployment & Scale-Up | | **Phase 5 Activities** | | 9.3.1 | Optimization & maintenance at scale | Phase 5 | 2043-10-01 | 2045-09-30 | 104 | Deployment & Scale-Up |

In [7]:

```
# %%
# Timeline: phases (background) + key activities (overlay)
import pandas as pd
import plotly.express as px

phase_rows = [
    {"Phase":"Phase 1 – Seed Deployment","Start":"2025-10-01","Finish":"2028-09-30"},
    {"Phase":"Phase 2 – Bootstrap Replication","Start":"2028-10-01","Finish":"2031-09-30"},
    {"Phase":"Phase 3 – Infrastructure Expansion","Start":"2031-10-01","Finish":"2037-09-30"},
    {"Phase":"Phase 4 – Dyson Swarm Assembly","Start":"2037-10-01","Finish":"2043-09-30"},
    {"Phase":"Phase 5 – Full Operational Capability","Start":"2043-10-01","Finish":"2045-09-30"},
]
activity_rows = [
    {"Task":"Program setup & ethics governance","Phase":"Phase 1 – Seed Deployment","Start":"2025-10-01","Finish":"2025-11-26"},
    {"Task":"Seed Gen-0 mechatronics","Phase":"Phase 1 – Seed Deployment","Start":"2025-11-27","Finish":"2026-04-14"},
    {"Task":"Autonomy v1 (payload-centric)","Phase":"Phase 1 – Seed Deployment","Start":"2026-01-15","Finish":"2026-06-30"},
    {"Task":"Env. testing (seed units)","Phase":"Phase 1 – Seed Deployment","Start":"2026-07-01","Finish":"2026-10-20"},
    {"Task":"Ground replication dry-run","Phase":"Phase 1 – Seed Deployment","Start":"2026-10-21","Finish":"2027-02-08"},
    {"Task":"Mass driver PDR","Phase":"Phase 1 – Seed Deployment","Start":"2026-10-21","Finish":"2027-04-04"},
    {"Task":"Pathfinder AIT/launch prep","Phase":"Phase 1 – Seed Deployment","Start":"2028-01-01","Finish":"2028-07-15"},

    {"Task":"Site selection & resource mapping","Phase":"Phase 2 – Bootstrap Replication","Start":"2028-10-01","Finish":"2029-03-31"},
    {"Task":"Mining + processing line v1","Phase":"Phase 2 – Bootstrap Replication","Start":"2029-04-01","Finish":"2030-03-31"},
    {"Task":"Additive mfg cell v1","Phase":"Phase 2 – Bootstrap Replication","Start":"2029-07-01","Finish":"2030-03-31"},

    {"Task":"Panel fab line (cluster 1)","Phase":"Phase 3 – Infrastructure Expansion","Start":"2031-10-01","Finish":"2032-07-07"},
    {"Task":"Mass driver prototype build & fire","Phase":"Phase 3 – Infrastructure Expansion","Start":"2032-01-01","Finish":"2032-10-08"},
    {"Task":"Power-beaming pilot","Phase":"Phase 3 – Infrastructure Expansion","Start":"2036-06-01","Finish":"2036-11-30"},

    {"Task":"Multi-cluster fab ramp","Phase":"Phase 4 – Dyson Swarm Assembly","Start":"2037-10-01","Finish":"2040-10-01"},
    {"Task":"Multi-driver network online","Phase":"Phase 4 – Dyson Swarm Assembly","Start":"2039-07-01","Finish":"2041-01-15"},

    {"Task":"Optimization & maintenance at scale","Phase":"Phase 5 – Full Operational Capability","Start":"2043-10-01","Finish":"2045-09-30"},
]

ph_df = pd.DataFrame(phase_rows)
ph_df['Start'] = pd.to_datetime(ph_df['Start'])
ph_df['Finish'] = pd.to_datetime(ph_df['Finish'])
act_df = pd.DataFrame(activity_rows)
act_df['Start'] = pd.to_datetime(act_df['Start'])
act_df['Finish'] = pd.to_datetime(act_df['Finish'])

fig_ph = px.timeline(ph_df, x_start="Start", x_end="Finish", y="Phase", color="Phase", title="Phase Roadmap (Gantt)")
fig_ph.update_yaxes(autorange="reversed")
fig_ph.update_traces(opacity=0.35)
fig_ph.show()

fig_act = px.timeline(act_df, x_start="Start", x_end="Finish", y="Phase", color="Task", title="Key Activities by Phase")
fig_act.update_yaxes(autorange="reversed")
fig_act.show()
```