3D modeling or printing specific parts of the ship.
Deliver realistic technical documentation as if you were part of an advanced aerospace team.

STRUCTURAL TECHNICAL DOCUMENTATION: ULTRA-RESISTANT WAREHOUSE

MODULAR GLOBAL STRUCTURE

> Inspiration: solar probes as*Parker Solar Probe*, type modules*Orion*, and type layers*Starship Heatshield*

a) Structural core (main frame)

- Material: Honeycomb-type alveolar panels made of CFRP or lightweight titanium.
- Design:
  - Configuration Hexcentric With attachable modules.
  - Structure reinforced by internal Ti-Al alloy supports with thermal spacers.
- Mounting:
  - Mechanical joints with self-centering bimetallic screws.
  - Reinforcement with friction stir welding (FSW) or laser sintering bonding.

b) Ensamblaje tipo plug-in / plug-lock

- Independent modules (sensors, cabin, shield, motor).
- Each module has mechanical + electrical + thermal connectors.
- Design oriented towards rapid replacement and A/B testing.

c) Insulating inner layer (inertial thermal buffer)

- Layers:

- Silica Aerofoam
- Encapsulated thermal gel core
- 3D carbon fiber mesh with internal channels

- Function: Structural insulation + heat dissipation by controlled internal convection.

d) Ablative ceramic exterior

- Tile-type panels with ceramic coating (ZrB_2, SiC).
- Adhesion with high temperature resins (reinforced polyamides or phenols).
- Coating applied by means ofthermal plasma spraying (TPS coating).

ADVANCED MATERIALS MANUFACTURING

Nano Titanium CFRP

- Process:
  - Powder metallurgy (PM) with laser sintering.
  - Injection of triaxial fibers into a matrix with isotropic pressure.
  - Thermal curing in autoclave (200–400°C, >6 bar).
- Final treatment:
  - CVD ablative coating.
  - Fracture analysis with X-ray spectroscopy.

Cerámica-carbono (C/C Ceramic Shield)

- Oriented woven carbon fiber preforms.
- Infiltration with ceramic resin (precursor polymer) → Pyrolysis.

- Compaction in a press at >1600°C with an inert atmosphere.
- Deposition of metallic outer layer (beryllium or hafnium) by PVD.

C) Reinforced graphene glass

- Multilayer graphene laminate on borosilicate glass or sapphire.
- Molecular bonding between layers using pulsed laser.
- Reinforced with invisible internal metal micro mesh (HUD style).

D) Superconductors (for magnetic shield)

- Epitaxial deposition on flexible substrates.
- Modular coil design (toroidal architecture).
- Dielectric encapsulation + cryogenic connection.

ASSEMBLY DIAGRAM – CONNECTION SCHEMATIC BETWEEN MODULES

You can submit it as:

- SVG with tags: using tools like Inkscape or Blender + Grease Pencil.
- Technical CAD drawing: isometric view + indications of materials and anchors.
- Table or JSON of connections:

```
{
"HEAT_SHIELD": ["MAIN_FUSELAGE"],
"MAIN_PROPULSION": ["MAIN_FUSELAGE", "INSULATION_AND_INTERNAL_LAYER"],
"COCKPIT_AND_CONTROL": ["MAIN_FUSELAGE", "FPGA_CONTROL_MODULE"],
"QUANTUM_SENSORS": ["CABIN_AND_CONTROL", "MAGNETIC_SHIELD"],
```

```
"POWER_PANEL": ["MAIN_FUSELAGE"],
"ANTENNAS_AND_COMMUNICATIONS": ["CABIN_AND_CONTROL",
"FPGA_CONTROL_MODULE"]
}
```

STRUCTURE AND MATERIAL FABRICATION – SUMMARY

- **Core Frame**: High-strength honeycomb titanium-CFRP, assembled with self-centering titanium bolts and laser sintering bonding.
- **Thermal Shield**: Layered Ceramic-Carbon-Carbon system, manufactured via carbon fiber preforms, pyrolysis, and PVD coating.
- **Quantum Sensor Dome**: Sapphire crystal grown via Czochralski method, layered with graphene sheets (layer-by-layer deposition).
- **Superconductive Shielding**: YBCO rings via epitaxial growth on flexible substrates, cryo-cooled and toroidally configured.

Assembly design:

- Modular interconnection via magnetic plug-locks.
- Vibration damping with PZT smart materials.
- Heat isolation via aerofoam and thermal gel.

Manufacturing techniques used:

- Powder metallurgy (PM)
- Additive Manufacturing (3D Printing with reinforced filaments)
- Chemical Vapor Deposition (CVD)
- Plasma spraying
- Epoxy-based aerospace adhesives

PYTHON

```
import bpy
```

```
import math

# Delete all
bpy.ops.object.select_all(action='SELECT')
bpy.ops.object.delete(use_global=False)

# Create fuselage
bpy.ops.mesh.primitive_cylinder_add(radius=1, depth=6, location=(0,
0, 0))
fuselage = bpy.context.object
fuselage.name = "Fuselage"

# Create front heat shield (more detailed)
bpy.ops.mesh.primitive_uv_sphere_add(radius=1.05, location=(0, 0,
3.1))
shield = bpy.context.object
shield.name = "Shield"
# Clipping with boolean
bpy.ops.mesh.primitive_cube_add(size=2, location=(0, 0, 2.8))
cutter = bpy.context.object
bpy.context.view_layer.objects.active = shield
bpy.ops.object.modifier_add(type='BOOLEAN')
shield.modifiers["Boolean"].object = cutter
shield.modifiers["Boolean"].operation = 'INTERSECT'
bpy.ops.object.modifier_apply(modifier="Boolean")
bpy.data.objects.remove(cutter)

# Subdivision to the shield for smoothing
bpy.ops.object.modifier_add(type='SUBSURF')
shield.modifiers["Subdivision"].levels = 2
bpy.ops.object.shade_smooth()

# Left solar panel (curved using subdivision)
bpy.ops.mesh.primitive_plane_add(size=2, location=(-2, 0, 0))
panel_left = bpy.context.object
panel_left.name = "Solar_Panel_Left"
panel_left.scale = (1, 2, 0.05)
bpy.ops.object.modifier_add(type='SUBSURF')
panel_left.modifiers["Subdivision"].levels = 2
bpy.ops.object.shade_smooth()

# Right solar panel
bpy.ops.mesh.primitive_plane_add(size=2, location=(2, 0, 0))
panel_right = bpy.context.object
panel_right.name = "Panel_Solar_Der"
panel_right.scale = (1, 2, 0.05)
bpy.ops.object.modifier_add(type='SUBSURF')
panel_right.modifiers["Subdivision"].levels = 2
bpy.ops.object.shade_smooth()

# Rear motors (cones)
```

```
for i in [-0.5, 0.5]:
    bpy.ops.mesh.primitive_cone_add(radius1=0.3, depth=0.6,
location=(i, 0, -3.3))
    engine = bpy.context.object
    engine.name = f"Motor_{i}"
engine.rotation_euler[0] = math.radians(180) # Points down

# Carbon-like material (basic nodes)
mat_carbon = bpy.data.materials.new(name="MaterialCarbono")
mat_carbon.use_nodes = True
nodes = mat_carbon.node_tree.nodes
bsdf = nodes.get("Principled BSDF")
if bsdf:
    bsdf.inputs["Base Color"].default_value = (0.1, 0.1, 0.1, 1)
    bsdf.inputs["Roughness"].default_value = 0.6
    bsdf.inputs["Metallic"].default_value = 0.4

# Apply materials
for obj in [fuselage, shield]:
    obj.data.materials.append(mat_carbon)

# Create collection
if "NaveEspacial" not in bpy.data.collections:
    collection = bpy.data.collections.new(name="NaveEspacial")
    bpy.context.scene.collection.children.link(collection)
else:
    collection = bpy.data.collections["NaveEspacial"]

for obj in [fuselage, shield, panel_left, panel_right]:
    collection.objects.link(obj)

# Add engines to the collection
for obj in bpy.context.scene.objects:
    if "Motor" in obj.name:
        collection.objects.link(obj)

# Simple takeoff animation
fuselage.location = (0, 0, 0)
fuselage.keyframe_insert(data_path="location", frame=1)
fuselage.location = (0, 0, 10)
fuselage.keyframe_insert(data_path="location", frame=50)

# Synchronize shield and panels with fuselage
for obj in [shield, panel_left, panel_right]:
    obj.parent = fuselage

# Also the engines
for obj in bpy.context.scene.objects:
    if "Motor" in obj.name:
        obj.parent = fuselage
```

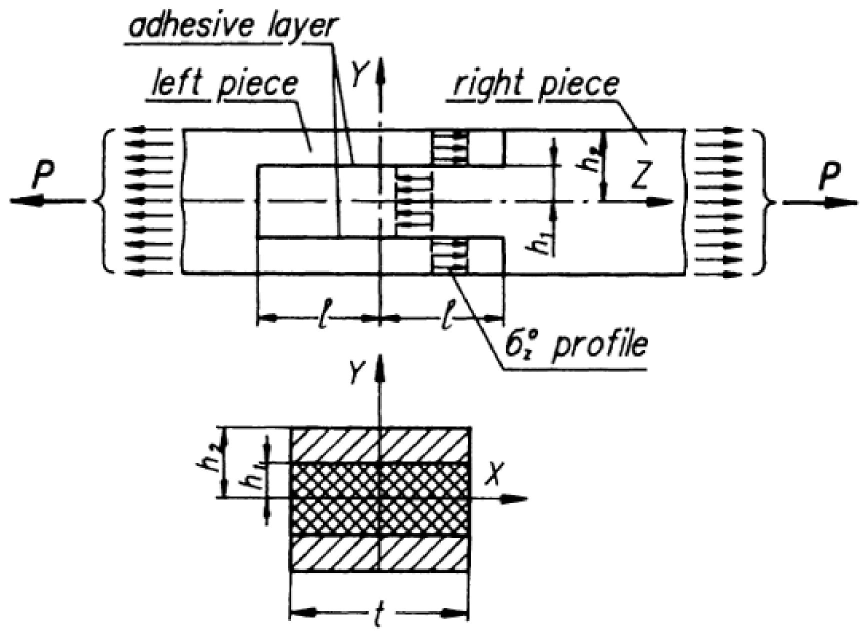

Figure 5.6 Loads applied to joint elements.