# 2025-Jun-09-Summary

```
# -*- coding: utf-8 -*-

# Final consolidation: regenerate clean outputs in 5 formats (JSON/CSV/PDF/Document/txt)

# and surface the Python artifacts in-room.

import json, hashlib, os, io, zipfile

from pathlib import Path

from datetime import datetime, timezone, timedelta

import pandas as pd

import matplotlib.pyplot as plt



from caas_jupyter_tools import display_dataframe_to_user



TZ = timezone(timedelta(hours=7))

BASE = Path("/mnt/data")

OUT = BASE / "FINAL_PACKAGE_KABUKI_INV_2025-06-09"

OUT.mkdir(exist_ok=True)



# -----------------------------

# 1) Canonical summary objects

# -----------------------------
```

```
case_id = "KABUKI-INV"
now = datetime.now(TZ).strftime("%Y-%m-%dT%H:%M:%S%z")
devices = ["iP11Pro","iP12mini-1","iP12mini-2","iPad","iP15P-Ghost","iP12-Ghost"]

# Facts we established
facts = {
    "date_scope_utc7": "2025-06-09",
    "baseband_disconnects_utc7": [
        "2025-06-09T09:40:50+07:00",
        "2025-06-09T22:28:33+07:00",
        "2025-06-09T23:50:15+07:00"
    ],
    "support_contacts_utc7": [
        "2025-06-09T13:42:00+07:00",
        "2025-06-09T13:44:00+07:00"
    ],
    "same_second_or_minute_match": False
}

hypotheses = [
    "Targeted interference linked to VN-Telco (hypothesis only; not proven)."
]


# Known outputs produced earlier (only include if files exist)
def fexists(path):
    return Path(path).exists()


produced = [
    {"type":"csv","name":"EVENTS.csv","path":"/mnt/data/KABUKI_INV_2025-06-09_OUT/EVENTS.csv"},
    {"type":"csv","name":"PIVOT.csv","path":"/mnt/data/KABUKI_INV_2025-06-09_OUT/PIVOT.csv"},
    {"type":"csv","name":"HUMAN_IMPACT_2025-06-09.csv","path":"/mnt/data/PHASE3_2025-06-09_HUMAN_IMPACT/HUMAN_IMPACT_2025-06-09.csv"},
    {"type":"csv","name":"JOIN_support_bb.csv","path":"/mnt/data/PHASE2_2025-06-09/JOIN_support_bb.csv"},
    {"type":"pdf","name":"SUMMARY.pdf","path":"/mnt/data/PHASE2_2025-06-09/SUMMARY.pdf"},
    {"type":"pdf","name":"HUMAN_IMPACT_2025-06-09.pdf","path":"/mnt/data/PHASE3_2025-06-09_HUMAN_IMPACT/HUMAN_IMPACT_2025-06-09.pdf"},
    {"type":"txt","name":"sha256_chain_generated.txt","path":"/mnt/data/ZIP_INTEGRATION_2025-06-09_OUT/sha256_chain_generated.txt"}
]
produced = [p for p in produced if fexists(p["path"])]


# -----------------------------
# 2) Build the 5 requested outputs
# ------------------------------

# (a) JSON
json_payload = {
    "case_id": case_id,
    "generated_at_utc7": now,
    "timezone": "UTC+7",
    "talkroom_status": "active",
    "devices_normalized": devices,
    "facts": facts,
    "hypotheses": hypotheses,
    "artifacts_found": produced
}
json_path = OUT / "final_summary.json"
json_path.write_text(json.dumps(json_payload, ensure_ascii=False, indent=2), encoding="utf-8")

# (b) CSV — flat digest table
rows = []
for p in produced:
    # compute sha256 if small enough; otherwise skip
    pth = Path(p["path"])
    sha = ""
    try:
        h = hashlib.sha256()
        with pth.open("rb") as f:
            for chunk in iter(lambda: f.read(1<<20), b""):
                h.update(chunk)
        sha = h.hexdigest()
    except Exception:
        sha = ""
    rows.append({
        "type": p["type"],
        "name": p["name"],
        "path": p["path"],
        "bytes": pth.stat().st_size if pth.exists() else "",
        "sha256": sha
    })

csv_df = pd.DataFrame(rows, columns=["type","name","path","bytes","sha256"])
csv_path = OUT / "final_digest.csv"
csv_df.to_csv(csv_path, index=False)
```

```
# (c) PDF — one-pager
pdf_path = OUT / "final_overview.pdf"
lines = []
lines.append("KABUKI-INV — Final Consolidation (Phase 1–4)")
lines.append(f"Generated (UTC+7): {now}")
lines.append("")
lines.append("Facts:")
lines.append(" ・Baseband disconnects: 09:40:50 / 22:28:33 / 23:50:15 (UTC+7)")
lines.append(" ・Apple Support contacts: 13:42 / 13:44 (UTC+7)")
lines.append(" ・No same-second/minute match; same-day sequence confirmed.")
lines.append("Hypotheses:")
lines.append(" ・Targeted interference linked to VN-Telco (hypothesis only; not proven).")
lines.append("Artifacts (subset):")
for r in rows[:8]:
   lines.append(f"  - {r['name']} ({r['sha256'][:12]}…)")
fig = plt.figure(figsize=(8.27, 11.69))
plt.text(0.06, 0.95, "¥n".join(lines), va="top", family="monospace")
plt.axis("off")
fig.savefig(pdf_path, format="pdf")
plt.close(fig)
```

```
# (d) Document — Markdown
md_path = OUT / "final_report.md"
md = f"""# KABUKI-INV — Final Consolidation (Phase 1–4)

- Generated (UTC+7): {now}
- Devices: {", ".join(devices)}

## Facts
- Date scope: 2025-06-09 (UTC+7)
- Baseband disconnects (iP12mini-1): 09:40:50 / 22:28:33 / 23:50:15
- Apple Support contacts (iP12-Ghost): 13:42 / 13:44
- No same-second/minute match; same-day sequence confirmed.

## Hypotheses (explicitly labeled)
- Targeted interference linked to VN-Telco (hypothesis only; not proven).

## Artifacts
{os.linesep.join([f"- {r['name']} — {r['sha256']}" for r in rows])}
"""
md_path.write_text(md, encoding="utf-8")
```

```
# (e) TXT — minimal checksum ledger

txt_path = OUT / "final_ledger.txt"

with txt_path.open("w", encoding="utf-8") as f:

    f.write("file¥tbytes¥tsha256¥tgenerated_at_utc7¥n")

    for r in rows:

        f.write(f"{r['name']}¥t{r['bytes']}¥t{r['sha256']}¥t{now}¥n")



# optional: a small ZIP with the five

zip_path = OUT / "final_bundle.zip"

with zipfile.ZipFile(zip_path, "w", compression=zipfile.ZIP_DEFLATED) as z:

    for p in [json_path, csv_path, pdf_path, md_path, txt_path]:

        z.write(p, arcname=p.name)



# Show digest in-room

display_dataframe_to_user("FINAL DIGEST (files & checksums)", csv_df)



{

    "json": json_path.as_posix(),

    "csv": csv_path.as_posix(),

    "pdf": pdf_path.as_posix(),
    "document_md": md_path.as_posix(),
    "txt": txt_path.as_posix(),
    "zip_all": zip_path.as_posix()
}
```

```
# Re-run after state reset (same code, compact)
import json, hashlib, os, io, zipfile
from pathlib import Path
from datetime import datetime, timezone, timedelta
import pandas as pd
import matplotlib.pyplot as plt
from caas_jupyter_tools import display_dataframe_to_user

TZ = timezone(timedelta(hours=7))
BASE = Path("/mnt/data")
OUT = BASE / "FINAL_PACKAGE_KABUKI_INV_2025-06-09"
OUT.mkdir(exist_ok=True)

case_id = "KABUKI-INV"
now = datetime.now(TZ).strftime("%Y-%m-%dT%H:%M:%S%z")
devices = ["iP11Pro","iP12mini-1","iP12mini-2","iPad","iP15P-Ghost","iP12-Ghost"]
facts = {
    "date_scope_utc7": "2025-06-09",
    "baseband_disconnects_utc7": ["2025-06-09T09:40:50+07:00","2025-06-09T22:28:33+07:00","2025-06-09T23:50:15+07:00"],
    "support_contacts_utc7": ["2025-06-09T13:42:00+07:00","2025-06-09T13:44:00+07:00"],
    "same_second_or_minute_match": False
}
hypotheses = ["Targeted interference linked to VN-Telco (hypothesis only; not proven)."]

def fexists(path): return Path(path).exists()
produced = [
    {"type":"csv","name":"EVENTS.csv","path":"/mnt/data/KABUKI_INV_2025-06-09_OUT/EVENTS.csv"},
    {"type":"csv","name":"PIVOT.csv","path":"/mnt/data/KABUKI_INV_2025-06-09_OUT/PIVOT.csv"},
    {"type":"csv","name":"HUMAN_IMPACT_2025-06-09.csv","path":"/mnt/data/PHASE3_2025-06-09_HUMAN_IMPACT/HUMAN_IMPACT_2025-06-09.csv"},
    {"type":"csv","name":"JOIN_support_bb.csv","path":"/mnt/data/PHASE2_2025-06-09/JOIN_support_bb.csv"},
    {"type":"pdf","name":"SUMMARY.pdf","path":"/mnt/data/PHASE2_2025-06-09/SUMMARY.pdf"},
    {"type":"pdf","name":"HUMAN_IMPACT_2025-06-09.pdf","path":"/mnt/data/PHASE3_2025-06-09_HUMAN_IMPACT/HUMAN_IMPACT_2025-06-09.pdf"},
    {"type":"txt","name":"sha256_chain_generated.txt","path":"/mnt/data/ZIP_INTEGRATION_2025-06-09_OUT/sha256_chain_generated.txt"}
]
produced = [p for p in produced if fexists(p["path"])]

json_payload = {"case_id":case_id,"generated_at_utc7":now,"timezone":"UTC+7","talkroom_status":"active",
"devices_normalized":devices,"facts":facts,"hypotheses":hypotheses,"artifacts_found":produced}

json_path = OUT / "final_summary.json"

json_path.write_text(json.dumps(json_payload, ensure_ascii=False, indent=2), encoding="utf-8")



rows = []

for p in produced:

    pth = Path(p["path"]); sha = ""

    try:

        h = hashlib.sha256()

        with pth.open("rb") as f:

            for chunk in iter(lambda: f.read(1<<20), b""):

                h.update(chunk)

        sha = h.hexdigest()

    except Exception: sha = ""

    rows.append({"type":p["type"],"name":p["name"],"path":p["path"],"bytes":pth.stat().st_size if
pth.exists() else "", "sha256":sha})

csv_df = pd.DataFrame(rows, columns=["type","name","path","bytes","sha256"])

csv_path = OUT / "final_digest.csv"

csv_df.to_csv(csv_path, index=False)



pdf_path = OUT / "final_overview.pdf"
```

```
lines = [
    "KABUKI-INV — Final Consolidation (Phase 1–4)",
    f"Generated (UTC+7): {now}",
    "",
    "Facts:",
    "  • Baseband disconnects: 09:40:50 / 22:28:33 / 23:50:15 (UTC+7)",
    "  • Apple Support contacts: 13:42 / 13:44 (UTC+7)",
    "  • No same-second/minute match; same-day sequence confirmed.",
    "Hypotheses:",
    "  • Targeted interference linked to VN-Telco (hypothesis only; not proven).",
    "Artifacts (subset):",
] + [f"  - {r['name']}  ({r['sha256'][:12]}…)" for r in rows[:8]]
fig = plt.figure(figsize=(8.27, 11.69))
plt.text(0.06, 0.95, "¥n".join(lines), va="top", family="monospace")
plt.axis("off")
fig.savefig(pdf_path, format="pdf")
plt.close(fig)

md_path = OUT / "final_report.md"
md = f"""# KABUKI-INV — Final Consolidation (Phase 1–4)
```

```
- Generated (UTC+7): {now}
- Devices: {", ".join(devices)}

## Facts
- Date scope: 2025-06-09 (UTC+7)
- Baseband disconnects (iP12mini-1): 09:40:50 / 22:28:33 / 23:50:15
- Apple Support contacts (iP12-Ghost): 13:42 / 13:44
- No same-second/minute match; same-day sequence confirmed.

## Hypotheses (explicitly labeled)
- Targeted interference linked to VN-Telco (hypothesis only; not proven).

## Artifacts
""" + os.linesep.join([f"- {r['name']} — {r['sha256']}" for r in rows])
md_path.write_text(md, encoding="utf-8")

txt_path = OUT / "final_ledger.txt"
with txt_path.open("w", encoding="utf-8") as f:
    f.write("file¥tbytes¥tsha256¥tgenerated_at_utc7¥n")
    for r in rows:
        f.write(f"{r['name']}¥t{r['bytes']}¥t{r['sha256']}¥t{now}¥n")
```

```
zip_path = OUT / "final_bundle.zip"

with zipfile.ZipFile(zip_path, "w", compression=zipfile.ZIP_DEFLATED) as z:

    for p in [json_path, csv_path, pdf_path, md_path, txt_path]:

        z.write(p, arcname=p.name)



display_dataframe_to_user("FINAL DIGEST (files & checksums)", csv_df)



{"json": json_path.as_posix(),

 "csv": csv_path.as_posix(),

 "pdf": pdf_path.as_posix(),

 "document_md": md_path.as_posix(),

 "txt": txt_path.as_posix(),

 "zip_all": zip_path.as_posix()}
```

### 結果

```
{'json': '/mnt/data/FINAL_PACKAGE_KABUKI_INV_2025-06-09/final_summary.json',
 'csv': '/mnt/data/FINAL_PACKAGE_KABUKI_INV_2025-06-09/final_digest.csv',
 'pdf': '/mnt/data/FINAL_PACKAGE_KABUKI_INV_2025-06-09/final_overview.pdf',
 'document_md': '/mnt/data/FINAL_PACKAGE_KABUKI_INV_2025-06-
09/final_report.md',
 'txt': '/mnt/data/FINAL_PACKAGE_KABUKI_INV_2025-06-09/final_ledger.txt',
 'zip_all': '/mnt/data/FINAL_PACKAGE_KABUKI_INV_2025-06-09/final_bundle.zip'}
```

### 5つの成果物（再生成）

- **JSON**: final_summary.json
    - フェーズ1–4の事実／仮説、生成物一覧、UTC+7の生成時刻を整理。
- **CSV**: final_digest.csv

  - 主要ファイルのtype,name,path,bytes,sha256ダイジェスト。
  - テーブルはこの部屋にも表示済み（checksums付き）。
- **PDF**: final_overview.pdf
  - 1ページ要約（Facts / Hypotheses / Artifacts digest）。
- **Document (Markdown)**: final_report.md
  - GitHubにそのまま貼れる最終レポ（事実は事実、仮説は仮説で明示）。
- **TXT（台帳）**：final_ledger.txt
  - ファイル名・サイズ・SHA256・生成時刻の台帳。
- まとめ **ZIP**（おまけ）：final_bundle.zip

### 中身の要点（事実／仮説を厳密に分離）

- **Facts（事実）**
  - 2025-06-09（UTC+7）に iPhone 12 mini-1 の baseband 切断が **09:40:50** / **22:28:33** / **23:50:15**。
  - 同日、iPhone 12-Ghost から Apple Support へ **13:42** / **13:44** に連絡。
  - **同秒・同分一致は無し**（same_second_or_minute_match=false）。
- **Hypothesis（仮説）**
  - 「VN-Telco 関連の標的干渉の可能性」— **未証明** として明記。