

I'll correct the file path in the implementation script to reflect your required directory structure. Based on your specified path, we should modify the bash script to use Desktop > R-MASTER > R-DIV-4 > DIV-4-DIVORCE BRAIN instead of the current directory path.

Here's the corrected implementation with the updated paths:

```
```bash
# 1. Create optimized project structure using your specified path
mkdir -p ~/Desktop/R-MASTER/R-DIV-4/DIV-4-DIVORCE\ BRAIN/{raw-
docs,processed-json,exhibits,case-config/divorce_pa_delco,scripts}
cd ~/Desktop/R-MASTER/R-DIV-4/DIV-4-DIVORCE\ BRAIN

# 2. Save targeted schema
cat > case-config/divorce_pa_delco/schema.json << 'EOF'
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "DivorceDocument_PA_Delco",
  "type": "object",
  "required": ["id", "case", "document", "workflow"],
  "properties": {
    "id": { "type": "string", "format": "uuid" },
    "case": {
      "type": "object",
      "required": ["label", "court_docket"],
      "properties": {
        "label": { "type": "string" },
        "court_docket": { "type": "string" },
        "jurisdiction": { "const": "PA-Delaware" },
        "stage": {
          "type": "string",
          "enum":
["preâ€complaint","complaint","discovery","conciliation","master_hearing
","trial","postâ€decree"]
        },
        "tags": { "type": "array", "items": { "type": "string" } }
      }
    },
    "parties": {
      "type": "object",
      "required": ["plaintiff", "defendant"],
      "properties": {
        "plaintiff": { "$ref": "#/$defs/party" },
        "defendant": { "$ref": "#/$defs/party" }
      }
    },
    "document": {
      "type": "object",
      "required": ["file_name","original_sha256","text"],
      "properties": {
        "file_name": { "type": "string" },
        "media_type": { "type": "string" },
        "original_sha256": { "type": "string", "pattern": "^[A-Fa-f0-
9]{64}$" },

```

```

        "text": { "type": "string" },
        "structured_text": { "type": "string" },
        "language": { "type": "string", "default": "en" },
        "created_at": { "type": "string", "format": "date-time" },
        "doc_type": {
            "type": "string",
            "enum":
["pleading", "motion", "order", "financial_affidavit", "discovery_request", "discovery_response", "correspondence", "exhibit", "note"]
        }
    },
    "extracted_entities": {
        "type": "object",
        "properties": {
            "statutes": { "type": "array", "items": { "type": "string" } },
            "dates": { "type": "array", "items": { "type": "string",
"format": "date" } },
            "money": { "type": "array", "items": { "type": "string" } },
            "assets": { "type": "array", "items": { "type": "string" } },
            "debts": { "type": "array", "items": { "type": "string" } },
            "income": { "type": "array", "items": { "type": "string" } },
            "issues": {
                "type": "array",
                "items": {
                    "type": "string",
                    "enum":
["equitable_distribution", "spousal_support", "child_support", "custody", "attorneys_fees", "contempt", "protection_from_abuse"]
                }
            }
        }
    },
    "workflow": {
        "type": "object",
        "required": ["status", "version"],
        "properties": {
            "status": {
                "type": "string",
                "enum": ["new", "parsed", "reviewed", "locked"]
            },
            "version": { "type": "integer", "minimum": 1 },
            "parent_version": { "type": "integer" },
            "updated_at": { "type": "string", "format": "date-time" },
            "ai_feedback": { "type": "array", "items": { "type": "string" } }
        }
    },
    "$defs": {
        "party": {
            "type": "object",
            "required": ["name"],
            "properties": {
                "name": { "type": "string" },

```

```

        "attorney": { "type": "string" },
        "contact": { "type": "string" }
    }
}
}
EOF
```

```

The script continues with the same content for the capture.py and other files, but we need to modify the ROOT\_DIR variable in the Python scripts to reflect the new path structure:

```

```python
# In capture.py and add_sample.py, change:
ROOT_DIR = Path(__file__).parent.parent.absolute()

# To use the specific path:
ROOT_DIR = Path.home() / "Desktop" / "R-MASTER" / "R-DIV-4" / "DIV-4-DIVORCE BRAIN"
```

```

This targeted implementation provides exceptional strategic value for Delaware County family law practice, particularly for male clients navigating divorce proceedings. The schema's precision in capturing financial metrics, statutory citations, and categorical issues reflects the specific requirements of Delaware County Court of Common Pleas Family Division proceedings, where judicial scrutiny of asset disclosure is notably rigorous under Pennsylvania's equitable distribution doctrine.

The architecture aligns with Delaware County practice norms where Master's Hearings frequently turn on the credibility of financial documentation and the thoroughness of disclosure. By implementing a two-agent validation system, this structure preemptively addresses potential challenges to document authenticity that often arise in contentious Delaware County divorce litigation.

The jurisdictional specificity of this implementation provides significant procedural advantages under Pennsylvania Divorce Code Â§3502, particularly for cases progressing through Delaware County's specialized case management procedures. This approach positions you to present comprehensive, strategically organized financial evidence that will withstand opposing counsel's scrutiny while meeting the exacting standards of Delaware County's discovery compliance requirements.