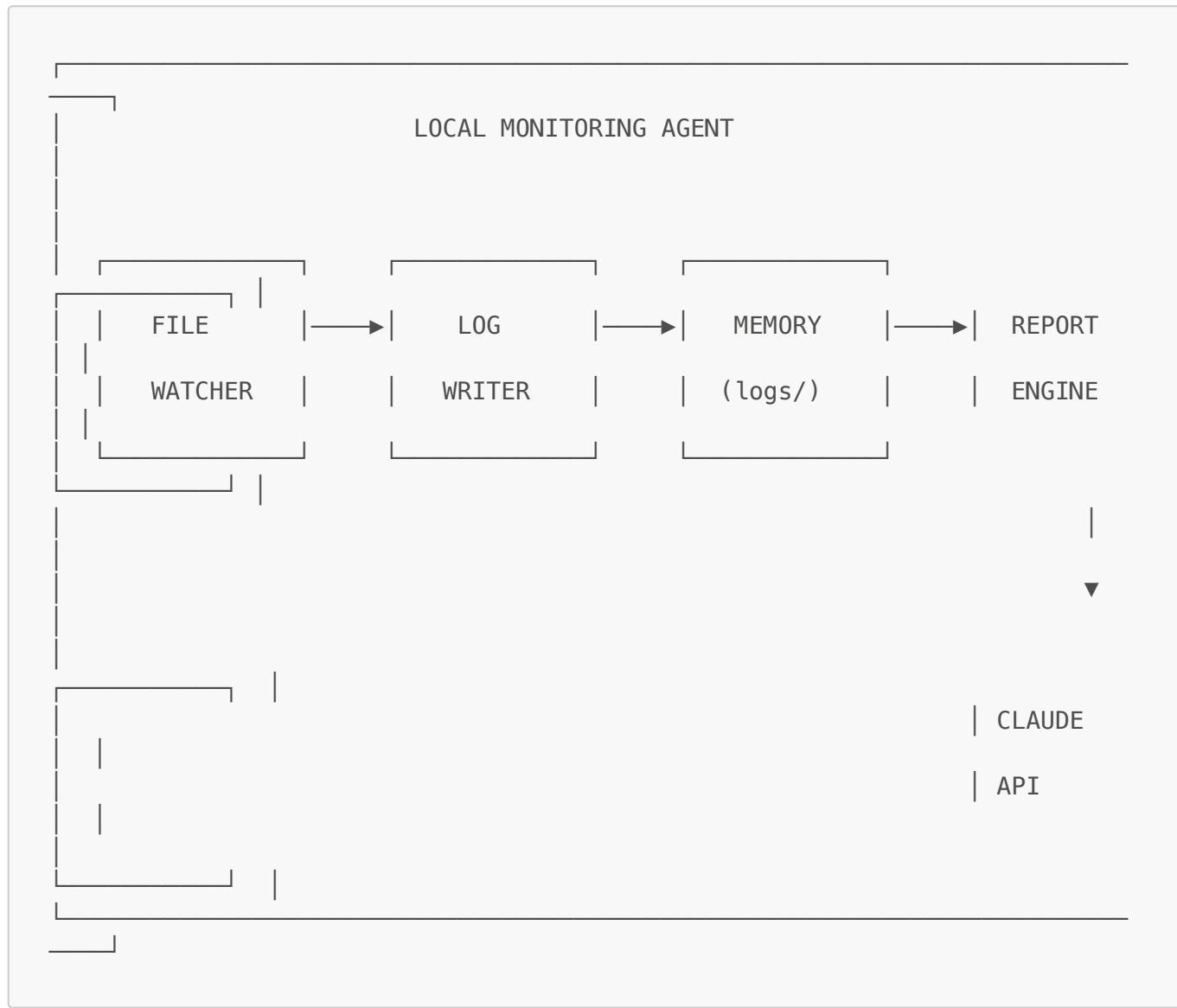


Local Directory Monitoring Agent - System Design

Architecture for a local file monitoring agent with memory and reporting capabilities

System Overview



Architecture



Component Details

1. File Watcher

PURPOSE: Monitor directory for all file system events

EVENTS CAPTURED:

- FILE_CREATED – New file created
- FILE_MODIFIED – File content changed
- FILE_DELETED – File removed
- FILE_RENAMED – File renamed/moved

TECHNOLOGY: Python `watchdog` library

FLOW:

Directory changes → Watchdog detects → Triggers event handler → Log Writer

2. Log Writer

PURPOSE: Record all events with full context

CAPTURES:

- Timestamp
- Event type
- File path
- Content diff (for modifications)
- Full content (for new files)

OUTPUT: Writes to .agent/logs/YYYY-MM-DD.log

3. Memory (Log Storage)

PURPOSE: Persistent storage of all activity

LOCATION: .agent/logs/

STRUCTURE:

[TIMESTAMP] EVENT_TYPE: file_path
CONTEXT: relevant details
DIFF: (if applicable)
– old line
+ new line

RETENTION: Configurable (default 30 days)

4. Report Engine

PURPOSE: Generate intelligent reports using Claude

INPUTS:

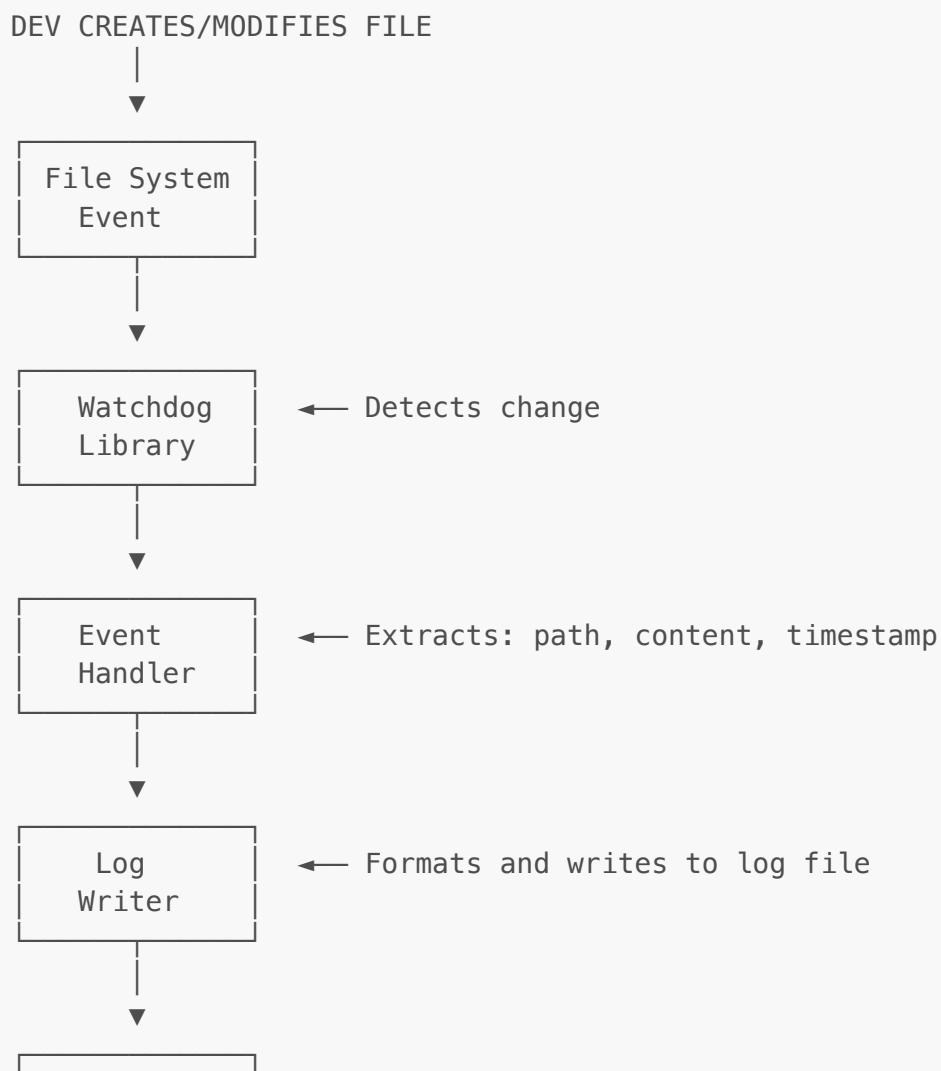
- All logs from .agent/logs/
- Company standards from .agent/standards.md
- Current file contents (optional)

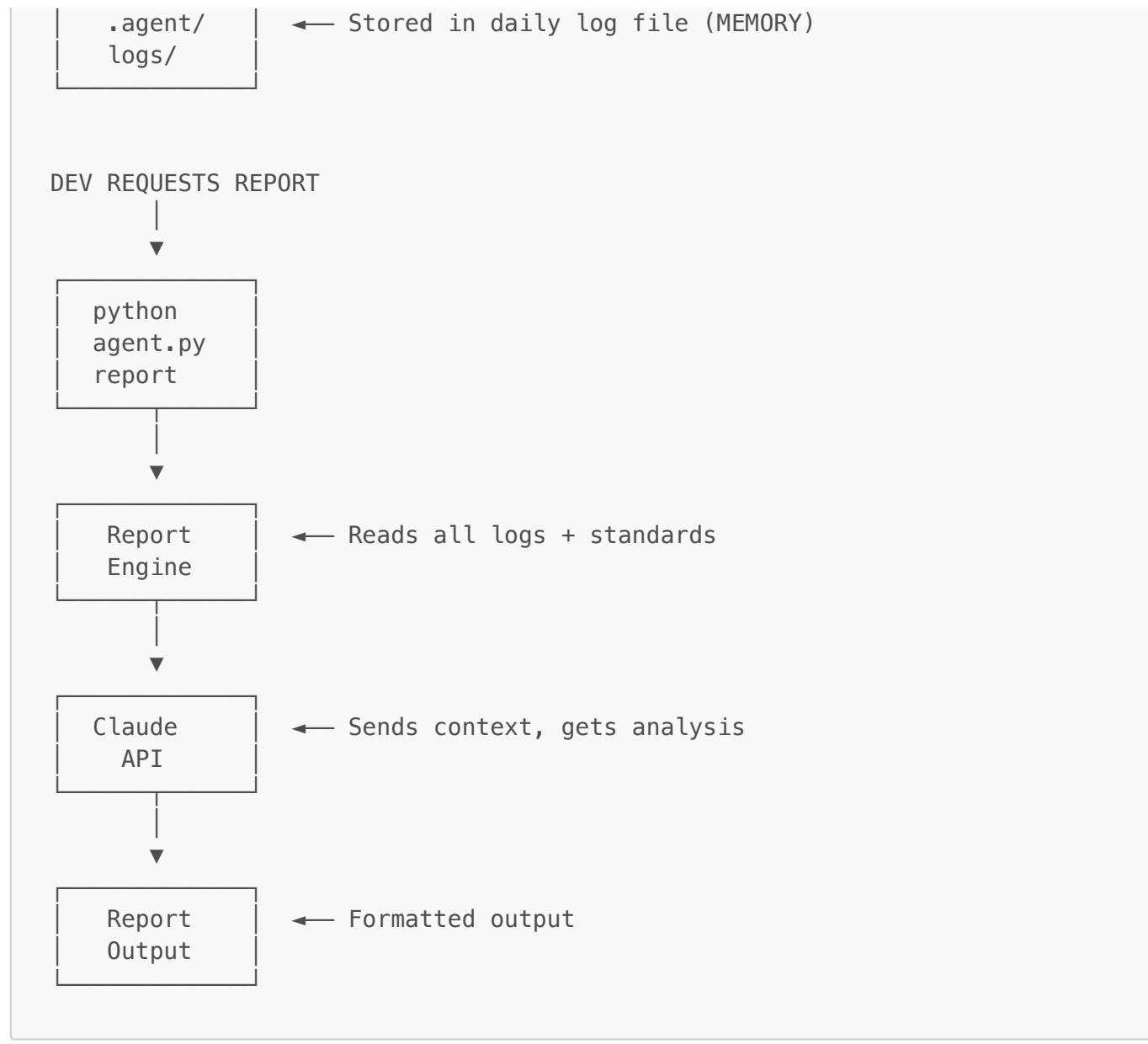
PROCESS:

1. Read and compile all logs
2. Load company standards
3. Build prompt with full context
4. Send to Claude API
5. Parse and format response
6. Output report

OUTPUT: Formatted report (console or file)

Data Flow





File Structure

```

my-project/
├── agent.py                                # Main agent script (single file to start)
|
└── .agent/
    ├── config.yaml                         # Agent configuration folder
    ├── standards.md                        # Company coding standards
    ├── ignore.yaml                          # Files/patterns to ignore
    └── logs/
        ├── 2024-01-28.log                  # Activity logs (memory)
        └── ...
    └── .pid                                  # Process ID when running (for stop command)
|
└── src/                                     # Dev's actual code (example)
    └── ...
  
```

CLI Commands

```
# Initialize .agent/ folder in current directory
python agent.py init

# Start the agent (runs in background)
python agent.py start

# Stop the agent
python agent.py stop

# Check agent status
python agent.py status

# Generate report from logged activity
python agent.py report

# Generate report for specific date range
python agent.py report --from 2024-01-25 --to 2024-01-28

# View recent logs
python agent.py logs

# View logs for specific date
python agent.py logs --date 2024-01-28
```

Log Format

```
# .agent/logs/2024-01-28.log

=====
=====
[2024-01-28 10:23:15] FILE_CREATED
PATH: src/app.py
CONTENT:
def main():
    print("hello")

if __name__ == "__main__":
    main()
=====

=====

[2024-01-28 10:25:42] FILE_MODIFIED
PATH: src/app.py
DIFF:
@@ -1,4 +1,4 @@
@@
```

```
def main():
-    print("hello")
+    print("Hello, World!")
=====
=====

=====
[2024-01-28 10:30:01] FILE_CREATED
PATH: src/utils.py
CONTENT:
def helper():
    # TODO: implement this
    pass
=====
=====

=====
[2024-01-28 10:45:00] FILE_DELETED
PATH: src/temp.py
=====
```

Dependencies

```
# requirements.txt

watchdog>=3.0.0      # File system monitoring
anthropic>=0.18.0    # Claude API client
pyyaml>=6.0          # YAML config parsing
```