

Diagram 0

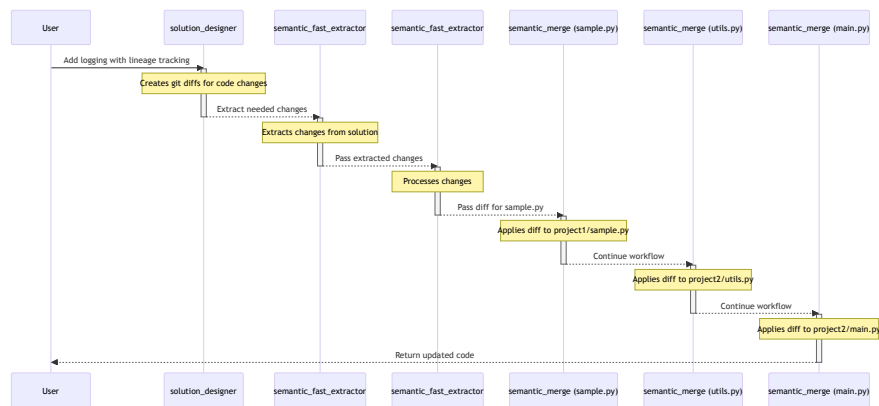


Diagram 1

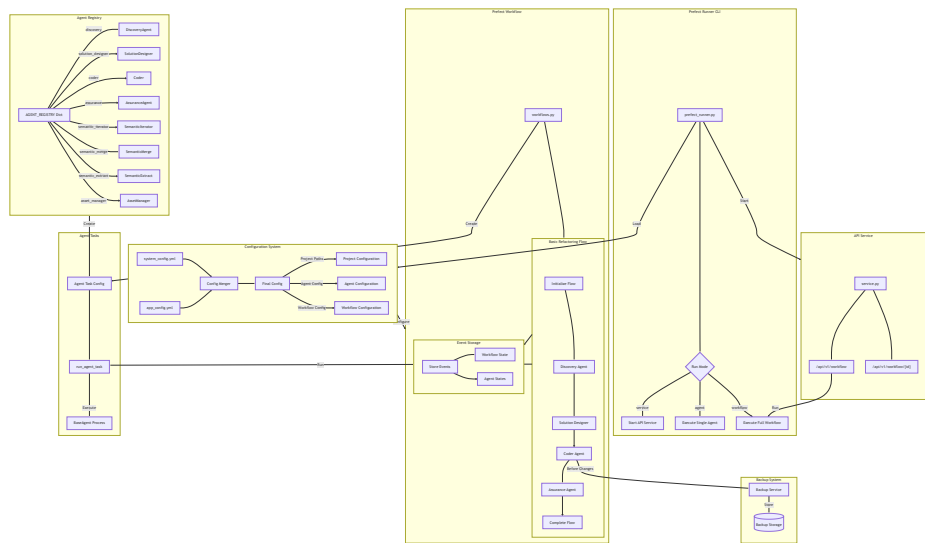


Diagram 2

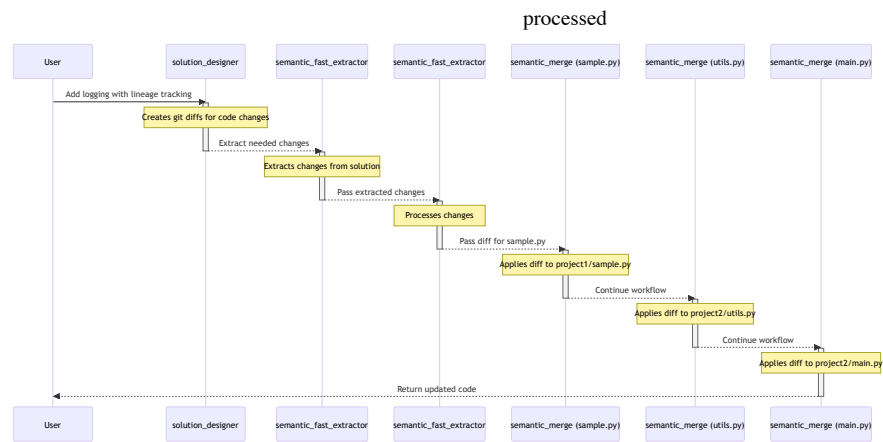


Diagram 3

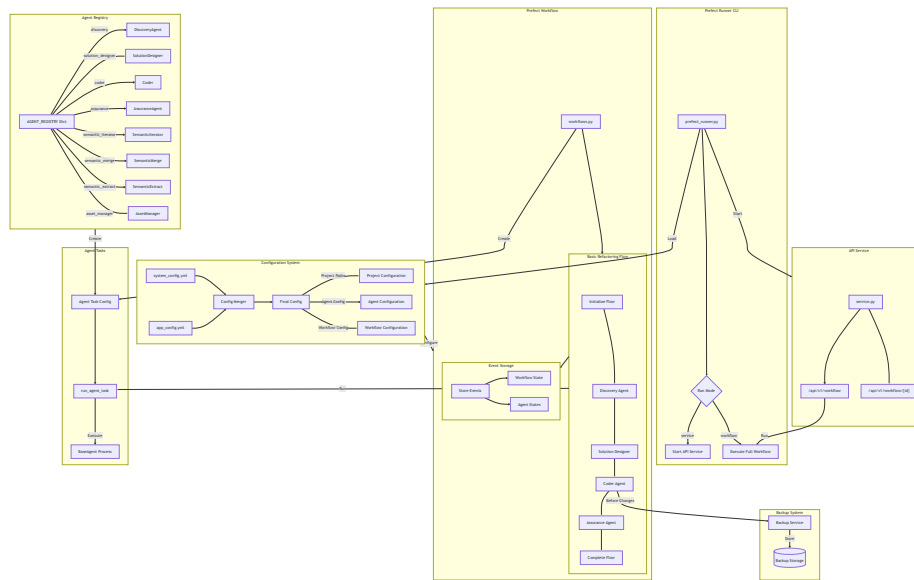


Diagram 4

flowchart TB

```
subgraph Config["Configuration System"]
    CF["system_config.yml<br>app_config.yml<br>user_config.yml"]
    CF --> CM[Config Merger]
    CM --> FC[Final Config]
    FC -->|Project Paths| PP[Project Configuration]
    FC -->|Agent Config| AConfig[Agent Configuration]
    FC -->|Workflow Config| WConfig[Workflow Configuration]
end
```

```
classDiagram
    %% Core Agent Base Classes
    class BaseAgent {
        +config: Dict
        +logger: Logger
        +__init__(config: Dict)
        +process(context: Dict) -> Dict
        #_get_agent_name() -> str
        #_get_agent_config() -> Dict
    }
```

```
}

%% Core Agents
class DiscoveryAgent {
    +process(context: Dict) -> Dict
    -_analyze_project() -> Dict
    -_extract_file_info() -> List
}

class SolutionDesigner {
    +process(context: Dict) -> Dict
    -_create_solution_plan() -> Dict
    -_generate_design() -> Dict
}

class Coder {
    +process(context: Dict) -> Dict
    -_implement_changes() -> List
    -_validate_changes() -> Bool
    -_backup_files() -> None
}

class AssuranceAgent {
    +process(context: Dict) -> Dict
    -_validate_implementation() -> Dict
    -_run_tests() -> Dict
}

%% Semantic Skills
class SemanticIterator {
    +config: Dict
    +instruction: str
    +format: str
    +__init__(config: Dict)
    +process(context: Dict) -> Dict
    -_iterate_items() -> List
    -_process_batch() -> List
    -_process_sequentially() -> List
}

class SemanticExtract {
    +process(context: Dict) -> Dict
    -_extract_data() -> List
}

class SemanticMerge {
    +process(context: Dict) -> Dict
    -_merge_content() -> Dict
}

class AssetManager {
```

```

        +process(context: Dict) -> Dict
        -_manage_assets() -> Dict
        -_track_assets() -> List
    }

%% Configuration Classes
class ConfigManager {
    +load(path: str) -> Dict
    +merge(base: Dict, override: Dict) -> Dict
    +locate_config(config: Dict, target: str) -> Dict
}

%% Project Classes
class Project {
    +path: Path
    +workspace: Path
    +__init__(path: Path, workspace: Path)
    +get_files() -> List
    +read_file(path: Path) -> str
    +write_file(path: Path, content: str) -> None
    +backup_file(path: Path) -> Path
}

%% Workflow Classes
class PrefectWorkflow {
    +run_basic_workflow(project_path: Path, intent_desc: Dict,
config: Dict) -> Dict
    -_initialize_flow() -> None
    -_configure_agents() -> Dict
    -_store_results() -> None
}

class AgentTaskConfig {
    +agent_class: Type
    +config: Dict
    +task_name: str
    +__init__(agent_class: Type, config: Dict, task_name: str)
}

%% API Service Classes
class APIService {
    +app: FastAPI
    +create_app(default_config: Dict) -> FastAPI
    -_register_routes() -> None
}

class WorkflowRequest {
    +project_path: str
    +intent: Dict
    +system_config: Dict
    +app_config: Dict

```

```

    }

    class WorkflowResponse {
        +workflow_id: str
        +status: str
        +storage_path: str
        +error: str
    }

    %% CLI Classes
    class PrefectRunner {
        +run_flow(mode: RunMode, config: Dict, agent_type: str) ->
Dict
        +load_configs(app_config_path: str, system_config_paths: List)
-> Dict
        +format_output(data: Dict, mode: RunMode) -> None
        +main() -> None
    }

    class RunMode {
        AGENT
        WORKFLOW
        SERVICE
    }

    %% Relationships
    BaseAgent <|-- DiscoveryAgent
    BaseAgent <|-- SolutionDesigner
    BaseAgent <|-- Coder
    BaseAgent <|-- AssuranceAgent

    BaseAgent <|-- SemanticIterator
    BaseAgent <|-- SemanticExtract
    BaseAgent <|-- SemanticMerge
    BaseAgent <|-- AssetManager

    PrefectWorkflow ..> AgentTaskConfig : creates
    PrefectWorkflow ..> BaseAgent : uses

    PrefectRunner ..> ConfigManager : uses
    PrefectRunner ..> PrefectWorkflow : executes
    PrefectRunner ..> APIService : starts

    APIService ..> WorkflowRequest : processes
    APIService ..> WorkflowResponse : returns
    APIService ..> PrefectWorkflow : executes

    PrefectWorkflow ..> Project : operates on
    Coder ..> Project : modifies
    DiscoveryAgent ..> Project : analyzes

```

