In [1]:
```python
# alice_bob_kqml_kif_demo.ipynb
# Agent communication using KQML with KIF content
# Scenario: Alice (procurement) queries Bob (warehouse) about 50-inch television

from textwrap import indent

# --- Helper functions ---

def kif_list(items):
    return " ".join(str(x) for x in items)

def kif_and(*forms):
    """Combine KIF expressions with (and ...)"""
    return f"(and\n{indent(chr(10).join(forms), '  ')}\n)"

def kqml(performative, **slots):
    """Render a simple KQML message with optional indented KIF content"""
    lines = [f"(kqmlmsg", f" :performative    {performative}"]
    for k, v in slots.items():
        if k == "content" and isinstance(v, str) and v.startswith("("):
            lines.append(f" :{k} \n {indent(v, ' ')}")
        else:
            lines.append(f" :{k} {v}")
    lines.append(")")
    return "\n".join(lines)
```

In [2]:
```python
# --- Mini ontology / warehouse data ---

WAREHOUSE = {
    "LG50UP8000":   {"type": "Television", "screen_inches": 50, "stock": 12, "hd
    "SAMS50AU7100": {"type": "Television", "screen_inches": 50, "stock": 7,  "hd
    "SONY50X80L":   {"type": "Television", "screen_inches": 50, "stock": 5,  "hd
    "LG55QNED80":   {"type": "Television", "screen_inches": 55, "stock": 9,  "hd
}

def total_stock_50in():
    """Compute total stock for 50-inch televisions."""
    return sum(m["stock"] for m in WAREHOUSE.values()
               if m["type"] == "Television" and m["screen_inches"] == 50)

def per_model_50in():
    """List model, stock, and HDMI ports for 50-inch televisions."""
    return [
        (mid, m["stock"], m["hdmi"])
        for mid, m in WAREHOUSE.items()
        if m["type"] == "Television" and m["screen_inches"] == 50
    ]
```

In [3]:
```python
# Alice asks Bob for total stock of 50-inch televisions
msg1 = kqml(
    "ask-one",
    sender="Alice",
    receiver="Bob",
    language="KIF",
    ontology="warehouse-ont-v1",
    reply_with="msg-001",
    content="(exists (?qty) (total-stock Television 50-inch ?qty))"
)
```

```
print(msg1, "\n")

# Bob replies with total stock
qty = total_stock_50in()
msg2_content = f"(= (total-stock Television 50-inch) {qty})"
msg2 = kqml(
    "tell",
    sender="Bob",
    receiver="Alice",
    language="KIF",
    ontology="warehouse-ont-v1",
    in_reply_to="msg-001",
    content=msg2_content
)
print(msg2)
```

```
(kqmlmsg
 :performative    ask-one
 :sender Alice
 :receiver Bob
 :language KIF
 :ontology warehouse-ont-v1
 :reply_with msg-001
 :content
  (exists (?qty) (total-stock Television 50-inch ?qty))
)

(kqmlmsg
 :performative    tell
 :sender Bob
 :receiver Alice
 :language KIF
 :ontology warehouse-ont-v1
 :in_reply_to msg-001
 :content
  (= (total-stock Television 50-inch) 24)
)
```

In [4]:

```
# Alice requests detailed model info
msg3 = kqml(
    "ask-all",
    sender="Alice",
    receiver="Bob",
    language="KIF",
    ontology="warehouse-ont-v1",
    reply_with="msg-002",
    content="(setofall (?m ?qty ?hdmi)\n"
            "  (and (model ?m Television)\n"
            "       (screen-size-inches ?m 50)\n"
            "       (stock-level ?m ?qty)\n"
            "       (has-hdmi-ports ?m ?hdmi)))"
)
print(msg3, "\n")

# Bob provides per-model stock and HDMI data
facts = []
for mid, qty_i, hdmi in per_model_50in():
    facts.extend([
        f"(model {mid} Television)",
        f"(screen-size-inches {mid} 50)",
```

```
        f"(stock-level {mid} {qty_i})",
        f"(has-hdmi-ports {mid} {hdmi})",
    ])
msg4_content = kif_and(*facts)
msg4 = kqml(
    "tell",
    sender="Bob",
    receiver="Alice",
    language="KIF",
    ontology="warehouse-ont-v1",
    in_reply_to="msg-002",
    content=msg4_content
)
print(msg4)
```

```
(kqmlmsg
 :performative    ask-all
 :sender Alice
 :receiver Bob
 :language KIF
 :ontology warehouse-ont-v1
 :reply_with msg-002
 :content
  (setofall (?m ?qty ?hdmi)
   (and (model ?m Television)
        (screen-size-inches ?m 50)
        (stock-level ?m ?qty)
        (has-hdmi-ports ?m ?hdmi)))
)

(kqmlmsg
 :performative    tell
 :sender Bob
 :receiver Alice
 :language KIF
 :ontology warehouse-ont-v1
 :in_reply_to msg-002
 :content
  (and
   (model LG50UP8000 Television)
   (screen-size-inches LG50UP8000 50)
   (stock-level LG50UP8000 12)
   (has-hdmi-ports LG50UP8000 3)
   (model SAMS50AU7100 Television)
   (screen-size-inches SAMS50AU7100 50)
   (stock-level SAMS50AU7100 7)
   (has-hdmi-ports SAMS50AU7100 2)
   (model SONY50X80L Television)
   (screen-size-inches SONY50X80L 50)
   (stock-level SONY50X80L 5)
   (has-hdmi-ports SONY50X80L 4)
 )
)
```

In [5]:
```
# Alice confirms the received information
msg5 = kqml(
    "confirm",
    sender="Alice",
    receiver="Bob",
    language="KIF",
```

```
    ontology="warehouse-ont-v1",
    reply_with="msg-003",
    content=kif_and(
        f"(= (total-stock Television 50-inch) {qty})",
        "(forall (?m ?q)\n"
        "  (if (and (model ?m Television) (screen-size-inches ?m 50) (stock-leve
        "      (>= ?q 0)))"
    )
)
print(msg5, "\n")

# Bob agrees
msg6 = kqml(
    "agree",
    sender="Bob",
    receiver="Alice",
    in_reply_to="msg-003"
)
print(msg6)
```

```
(kqmlmsg
 :performative    confirm
 :sender Alice
 :receiver Bob
 :language KIF
 :ontology warehouse-ont-v1
 :reply_with msg-003
 :content
  (and
   (= (total-stock Television 50-inch) 24)
   (forall (?m ?q)
     (if (and (model ?m Television) (screen-size-inches ?m 50) (stock-level ?m ?
q))
         (>= ?q 0)))
 )
)

(kqmlmsg
 :performative    agree
 :sender Bob
 :receiver Alice
 :in_reply_to msg-003
)
```

In [ ]: