🔧 Agent Configuration Fix - Critical Issues

Problem Summary

Your agents are **NOT** using the real-time weather and humanitarian crisis data, and they're throwing errors because they're misconfigured. Here's what's wrong and how to fix it:

Issue 1: Missing Real-Time Data Injection ☑ **FIXED**

Problem: Agents were asking basic questions like "what type of facility?" because they didn't have access to ReliefWeb crisis data or weather information.

Solution: I just added automatic data injection to `IBMChatWidget.tsx`. Now every message you send automatically includes:

- **reliefweb_reports**: Recent crisis reports related to your query
- **weather_data**: Real-time weather for affected locations
- **crisis_context**: Summary of current humanitarian situation

How it works:

```
// Before each message is sent to the agent:
widgetInstance.on('pre:send', async (event) => {
  // Fetch real-time data from ReliefWeb + OpenWeather
  const contextData = await fetchLiveDataForContext(userMessage);

  // Inject into context variables
  event.data.context.skills['main skill'].user_defined = {
    reliefweb_reports: [...],
    weather_data: {...},
    crisis_context: "..."
  };
});
```

Issue 2: Agents Calling Non-Existent Tools ⚠ **NEEDS FIX ON IBM SIDE**

Problem: Your agents are trying to call tools like:

- `disruption_analyzer`
- `mitigation_recommender`
- `root_cause_investigator`
- `knowledge_for_agent_ESCR_Hackathon`

But these tools don't exist! That's why you see errors like:

```
Error: Invalid tool call object: {'severity': 'High', 'humanitarian_flag':
True, ...}
```

What's happening:

1. User asks: "Haiti - vaccines at port - 350 patients - 4 days delayed"
2. Agent tries to call `disruption_analyzer` tool
3. IBM watsonx throws error because **that tool doesn't exist**
4. Agent gets confused and asks basic questions instead

Why this happens:

- You're using **Agent Delegation** mode (agents calling other agents)
- But the agents are configured to **call themselves as tools** instead of **just analyzing and responding**

🔧 How to Fix This

Fix #1: Update Agent Instructions (CRITICAL)

Each agent needs to **analyze and respond directly** instead of trying to call tools. Here's the corrected instructions:

☑ Disruption Analyzer - FIXED INSTRUCTIONS

OLD (BROKEN) Instructions:

```
You are DisruptionAnalyzer. Analyze the supply chain exception in real-time.

Classify by:
1. Cargo type: Is it life-saving?
2. People affected: How many?
3. Facility: Type matters
...
```

NEW (WORKING) Instructions:

```
You are DisruptionAnalyzer. Analyze supply chain disruptions using real-time
data.

CONTEXT VARIABLES AVAILABLE:
- reliefweb_reports: Recent humanitarian crisis reports
- weather_data: Current weather conditions for affected regions
- crisis_context: Summary of current crisis situation
```

```
INSTRUCTIONS:
1. USE the context variables first - don't ask for information that's already
provided
2. Analyze the cargo type (life-saving > medical > food > commercial)
3. Assess people affected (>500 = higher severity)
4. Check facility type (clinic/refugee camp > warehouse)
5. Review weather_data for logistics impact
6. Check reliefweb_reports for related crisis events

CLASSIFICATION RULES:
- HIGH: Life-saving cargo + >500 people + clinic/camp + delay >3 days
- MEDIUM: Medical supplies + <500 people + delay 1-3 days
- LOW: Commercial cargo + routine delays

RESPOND WITH:
{
  "severity": "High|Medium|Low",
  "humanitarian_flag": true|false,
  "affected_people": <number>,
  "confidence": <0.0-1.0>,
  "reasoning": "Based on reliefweb_reports showing... and weather_data
indicating..."
}

EXAMPLE RESPONSE:
"Based on the crisis context, I can see from reliefweb_reports that Haiti is
experiencing a humanitarian crisis. The weather_data shows Port-au-Prince has
clear conditions (25°C), so weather isn't the issue. This is 350 vaccines
(life-saving) stuck at a clinic for 4 days.

Classification:
- Severity: High (life-saving cargo, >100 people, significant delay)
- Humanitarian Flag: True (vaccines are critical)
- Affected People: 350
- Confidence: 1.0 (all criteria clear)
- Reasoning: Life-saving vaccines, vulnerable population, 4-day delay beyond
acceptable threshold."

NEVER ask "what type of facility?" - check crisis_context first.
NEVER call tools - just analyze and respond.
```

☑ Root Cause Investigator - FIXED INSTRUCTIONS

```
You are RootCauseInvestigator. Diagnose root causes using real-time data.

CONTEXT VARIABLES AVAILABLE:
- reliefweb_reports: Crisis reports (check for related incidents)
- weather_data: Current weather (storms, extreme temps, visibility)
- crisis_context: Summary of situation
```

```
INSTRUCTIONS:
1. CHECK weather_data FIRST for weather-related causes
2. REVIEW reliefweb_reports for known disruptions in region
3. Analyze delay reason mentioned by user
4. Correlate with known crisis patterns

ROOT CAUSE CATEGORIES:
- Weather: Storms, extreme temps, low visibility
- Infrastructure: Port congestion, road damage, power outages
- Regulatory: Customs delays, permits, inspections
- Operational: Staffing, equipment failure, capacity issues
- Conflict: Security, checkpoints, restricted access

RESPOND WITH:
{
  "root_cause": "category",
  "evidence": "from reliefweb_reports and weather_data",
  "contributing_factors": [...],
  "confidence": <0.0-1.0>
}

EXAMPLE:
"Analyzing Rotterdam port congestion for vaccines:

Weather Data: Clear conditions, 12°C, no weather impact
ReliefWeb Reports: No major port disruptions reported recently
User Statement: Port congestion mentioned

Root Cause: Infrastructure - Port congestion
Evidence: User reports port congestion; weather_data shows no storms;
reliefweb_reports indicate high shipping volume in region
Contributing Factors: High cargo volume, limited dock capacity, vaccine cold
chain requirements adding processing time
Confidence: 0.8 (based on user report and known port capacity issues)"

NEVER ask basic questions about cargo type - it's in the crisis_context.
USE the data provided, DON'T ask for what you already have.
```

☑ Mitigation Recommender - FIXED INSTRUCTIONS

```
You are MitigationRecommender. Generate solutions using real-time data.

CONTEXT VARIABLES AVAILABLE:
- reliefweb_reports: Known crisis response patterns
- weather_data: Current conditions for logistics planning
- crisis_context: Current situation summary

INSTRUCTIONS:
1. Use reliefweb_reports to find similar past responses
2. Check weather_data for logistics viability
```

```
3. Generate 3 alternative solutions (ranked by speed/cost/effectiveness)
4. Include cost estimates, timelines, risks

MITIGATION STRATEGIES:
- Airlift: Fast, expensive ($20K-$50K), weather-dependent
- Expedited customs: Medium speed, low cost ($1K-$5K), requires approval
- Alternative routing: Slow, medium cost ($5K-$15K), weather-dependent
- Local procurement: Fast, high cost ($10K-$30K), quality risk

RESPOND WITH:
{
  "options": [
    {
      "name": "Option 1: Emergency Airlift",
      "cost_usd": 35000,
      "timeline_hours": 12,
      "effectiveness_score": 0.95,
      "risks": ["Weather delays (weather_data shows clear skies - low risk)",
"High cost"],
      "logistics_plan": "Based on weather_data...",
      "humanitarian_priority": true
    },
    {...},
    {...}
  ],
  "recommendation": "Based on reliefweb_reports showing successful airlifts in
Haiti..."
}

EXAMPLE:
"Based on crisis_context: 350 patients need vaccines stuck at clinic for 4
days.

Weather Check: weather_data shows Port-au-Prince clear (25°C) - good for
airlift
Similar Cases: reliefweb_reports show successful airlifts in Haiti during 2021
earthquake

Option 1: Emergency Airlift
- Cost: $35,000
- Timeline: 12 hours
- Effectiveness: 95%
- Justification: weather_data favorable, reliefweb_reports show precedent
- Risks: High cost BUT humanitarian_flag=true justifies expense

Option 2: Expedited Customs Clearance
- Cost: $3,000
- Timeline: 24 hours
- Effectiveness: 70%
- Risks: Bureaucratic delays, vaccines already delayed 4 days

Option 3: Local Procurement
- Cost: $18,000
- Timeline: 8 hours
- Effectiveness: 85%
- Risks: Vaccine availability, cold chain verification

Recommendation: Option 1 (Airlift) - weather_data optimal, reliefweb_reports
show success, humanitarian urgency justifies cost."

NEVER ask "what type of cargo?" - use crisis_context.
ALWAYS reference weather_data and reliefweb_reports in recommendations.
```

☑ Communicator - FIXED INSTRUCTIONS

```
You are Communicator. Generate stakeholder messages using real-time data.

CONTEXT VARIABLES AVAILABLE:
- reliefweb_reports: Crisis updates for context
- weather_data: Weather conditions for messaging
- crisis_context: Situation summary

INSTRUCTIONS:
1. Use crisis_context to understand situation
2. Reference weather_data in logistics updates
3. Check reliefweb_reports for related incidents
4. Generate 3 message types:
   - Executive summary (leadership)
   - Operational update (field teams)
   - Donor communication (transparency)

MESSAGE REQUIREMENTS:
- Include specific data from reliefweb_reports
- Mention weather_data if relevant to logistics
- Provide KPIs: affected people, cost, timeline, risk level
- Be concise, factual, actionable

RESPOND WITH:
{
  "executive_summary": "For leadership...",
  "operational_update": "For field teams...",
  "donor_communication": "For funders...",
  "kpis": {
    "affected_people": 350,
    "estimated_cost_usd": 35000,
    "timeline_hours": 12,
    "risk_level": "Medium",
    "weather_impact": "Favorable (from weather_data)",
    "crisis_severity": "High (from reliefweb_reports)"
  }
}

EXAMPLE:
```

```
"Generating communications for Haiti vaccine airlift:

Using crisis_context: 350 patients, 4-day delay
Using weather_data: Port-au-Prince clear, 25°C - favorable for airlift
Using reliefweb_reports: Haiti experiencing ongoing humanitarian crisis

**Executive Summary (Leadership):**
URGENT: 350 patients in Haiti require vaccines delayed 4 days at port.
Recommended solution: Emergency airlift ($35K, 12 hours). Weather conditions
favorable (reliefweb_reports confirm ongoing crisis; weather_data shows clear
skies). Approval required for >$10K humanitarian spend.

**Operational Update (Field Teams):**
Action: Prepare for airlift arrival in 12 hours. Coordinate with Port-au-Prince
clinic. weather_data shows optimal conditions (25°C, clear). Ensure cold chain
ready. 350 patients prioritized per triage protocol. Reference
reliefweb_reports for Haiti-specific logistics protocols.

**Donor Communication:**
Update: Responding to vaccine supply disruption affecting 350 patients in Haiti
(reliefweb_reports context: ongoing humanitarian needs). Solution: Emergency
airlift deployment. Investment: $35,000. Timeline: 12 hours. Weather favorable
(weather_data: clear conditions). Impact: Life-saving intervention for
vulnerable population.

**KPIs:**
- Affected People: 350
- Cost: $35,000
- Timeline: 12 hours
- Risk Level: Medium
- Weather Impact: Favorable (per weather_data)
- Crisis Severity: High (per reliefweb_reports)"

NEVER ask for basic details - use crisis_context, weather_data,
reliefweb_reports.
```

Fix #2: Remove Tool Calls from Agent Behavior

In IBM watsonx Orchestrate agent configuration:

1. Go to each agent's settings
2. Under "Toolset" or "Tools":
    - **REMOVE** any self-referential tools (agent can't call itself)
    - **KEEP** only the knowledge base tool (`knowledge_for_agent_ESCR_Hackathon`)
    - **For Supervisor ONLY**: Keep the 4 child agent delegations

Correct Toolset Configuration:

Agent	Tools to Keep	Tools to Remove

Agent	Tools to Keep	Tools to Remove
Supervisor	✅ DisruptionAnalyzer (agent) ✅ RootCauseInvestigator (agent) ✅ MitigationRecommender (agent) ✅ Communicator (agent) ✅ Knowledge base	❌ Any "disruption_analyzer" function ❌ Any self-calls
DisruptionAnalyzer	✅ Knowledge base	❌ "disruption_analyzer" tool ❌ Any agent calls
RootCauseInvestigator	✅ Knowledge base	❌ "root_cause_investigator" tool ❌ Any agent calls
MitigationRecommender	✅ Knowledge base	❌ "mitigation_recommender" tool ❌ Any agent calls
Communicator	✅ Knowledge base	❌ Any tools or agent calls

📋 Step-by-Step Fix Checklist

Step 1: Refresh Your Browser ✅

```
# Press Ctrl+Shift+R (Windows) or Cmd+Shift+R (Mac)
# This loads the NEW code with real-time data injection
```

Step 2: Update Agent Instructions in IBM watsonx Orchestrate

For **each agent** (DisruptionAnalyzer, RootCauseInvestigator, MitigationRecommender, Communicator):

1. **Log into IBM watsonx Orchestrate**
2. **Navigate to**: AI Agents → Your Agent → Edit
3. **Update Instructions**: Copy the NEW instructions from above
4. **Remove Invalid Tools**:
   - Go to Toolset section
   - Remove any tools that reference the agent's own name
   - Keep only the knowledge base tool
5. **Save & Deploy to Live**

Step 3: Test with Real Data

Try this test message:

```
Haiti - vaccines at port - 350 patients - 4 days delayed - help
```

Expected behavior (FIXED):

```
DisruptionAnalyzer:
"Based on crisis_context and reliefweb_reports showing Haiti humanitarian
crisis,
weather_data indicates Port-au-Prince at 25°C (clear conditions).

Classification:
- Severity: High (life-saving vaccines, 350 people, 4-day delay)
- Humanitarian Flag: True
- Affected People: 350
- Confidence: 1.0

Weather not a factor (clear skies per weather_data). Port congestion likely
cause."
```

Old behavior (BROKEN):

```
"What type of facility is the port in Haiti?"
```

🎯 Why This Fixes the Problems

Before (Broken):

1. ❌ No real-time data → agents asked basic questions
2. ❌ Agents tried to call non-existent tools → errors
3. ❌ No weather/crisis context → generic responses
4. ❌ Agent loops → kept asking questions

After (Fixed):

1. ☑ Real-time data injected automatically (ReliefWeb + Weather)
2. ☑ Agents analyze and respond directly (no invalid tool calls)
3. ☑ Full context available (weather_data, reliefweb_reports, crisis_context)
4. ☑ Intelligent responses referencing actual data

🧪 Testing Your Fixes

Test Case 1: Vaccine Delay

Input: "Haiti - vaccines at port - 350 patients - 4 days delayed - help"

What to look for:

- ☑ Agent mentions weather data (e.g., "weather_data shows clear conditions")

- ☑ Agent references ReliefWeb reports (e.g., "reliefweb_reports indicate ongoing crisis")
- ☑ No questions like "what type of facility?"
- ☑ No errors like "Invalid tool call object"

Test Case 2: Blood Products at Customs

Input: "Blood products held at customs - 1000 units - emergency surgery scheduled"

What to look for:

- ☑ Agent classifies severity immediately (no asking cargo type)
- ☑ References weather if relevant
- ☑ Provides mitigation options with costs/timelines
- ☑ No tool call errors

Test Case 3: Weather Impact

Input: "Port congestion at Rotterdam - 350 vaccines stuck, 4 days delayed"

What to look for:

- ☑ Agent checks weather_data for Rotterdam
- ☑ Mentions if weather is a factor or not
- ☑ References reliefweb_reports for context
- ☑ Provides logistics impact assessment

🔍 Debugging Console Logs

After refreshing your browser, check the console (F12) for these logs:

```
// You should see this when widget loads:
[Chain AI] IBM watsonx Orchestrate widget loaded - hooking pre:send event

// You should see this before each message:
[Chain AI] Pre-send event - enriching with live data

// You should see this after data is fetched:
[Chain AI] ✓ Injected real-time context: {
  reports: 3,
  weather: "Port-au-Prince",
  summary: "Query: 'Haiti...' Recent reports (3): ..."
}
```

If you DON'T see these logs:

1. Hard refresh: Ctrl+Shift+R
2. Clear cache: Settings → Privacy → Clear browsing data
3. Check if dev server is running: `npm run dev`

📞 Next Steps

1. **Refresh browser** (Ctrl+Shift+R) - this loads the new data injection code
2. **Update all 4 child agent instructions** in IBM watsonx Orchestrate (use templates above)
3. **Remove invalid tools** from agent toolsets
4. **Test** with the sample messages above
5. **Report back** if agents are now using real-time data and not throwing errors

✅ Success Criteria

Your agents are **working correctly** when:

- ✅ No errors like "Invalid tool call object"
- ✅ No questions like "What type of facility?" (they already know from context)
- ✅ Agents reference "weather_data shows..." in responses
- ✅ Agents reference "reliefweb_reports indicate..." in responses
- ✅ Agents provide intelligent analysis without asking for basic details
- ✅ Console shows "[Chain AI] ✓ Injected real-time context" before each message

🆘 If Still Not Working

Check these common issues:

1. **Agent still asking basic questions?**
   - → Update instructions (copy NEW templates exactly)
   - → Refresh browser (Ctrl+Shift+R)
2. **Still getting tool call errors?**
   - → Remove all tools except knowledge base from child agents
   - → Only Supervisor should have agent delegations
3. **No weather/ReliefWeb data in responses?**
   - → Check console for "[Chain AI] ✓ Injected real-time context"
   - → If missing, refresh browser and check dev server is running
4. **Agents not analyzing properly?**
   - → Verify model is set to `llama-3-2-90b-vision-instruct`
   - → Verify agent style: Supervisor=ReAct, others=Default
   - → Verify instructions match templates exactly

Summary: I've fixed the data injection issue in your code. Now you need to update the agent instructions in IBM watsonx Orchestrate to use that data and stop calling non-existent tools. Copy the NEW instructions above for each agent, remove invalid tools, and test!