

Verteiltes Monitoring

23. Oktober 2014



Mathias Kettner GmbH
Linux und Open Source



- Szenarien
- Entscheidungskriterien
- Best practices
- Was wir nicht verfolgen



Szenarien

- Mehrere Rechenzentren weltweit
- Überwachung tausender Märkte
- Überwachung als Managed Service
- Netztrennungen (DMZ)
- Aufteilung von Produktion / Integration
- Lastverteilungen
- Kleine Standorte mit 2-3 Geräten
- Dienste / Server im Internet



Entscheidungskriterien

- Sicherheit, wer darf auf wen zugreifen
- Zuverlässigkeit
- Netzwerkverkehr
- Konfigurationsaufwand
- Ist der Standort autonom?
- Skalierbarkeit
- Konsistenz der Daten



Best practice mit Check_MK

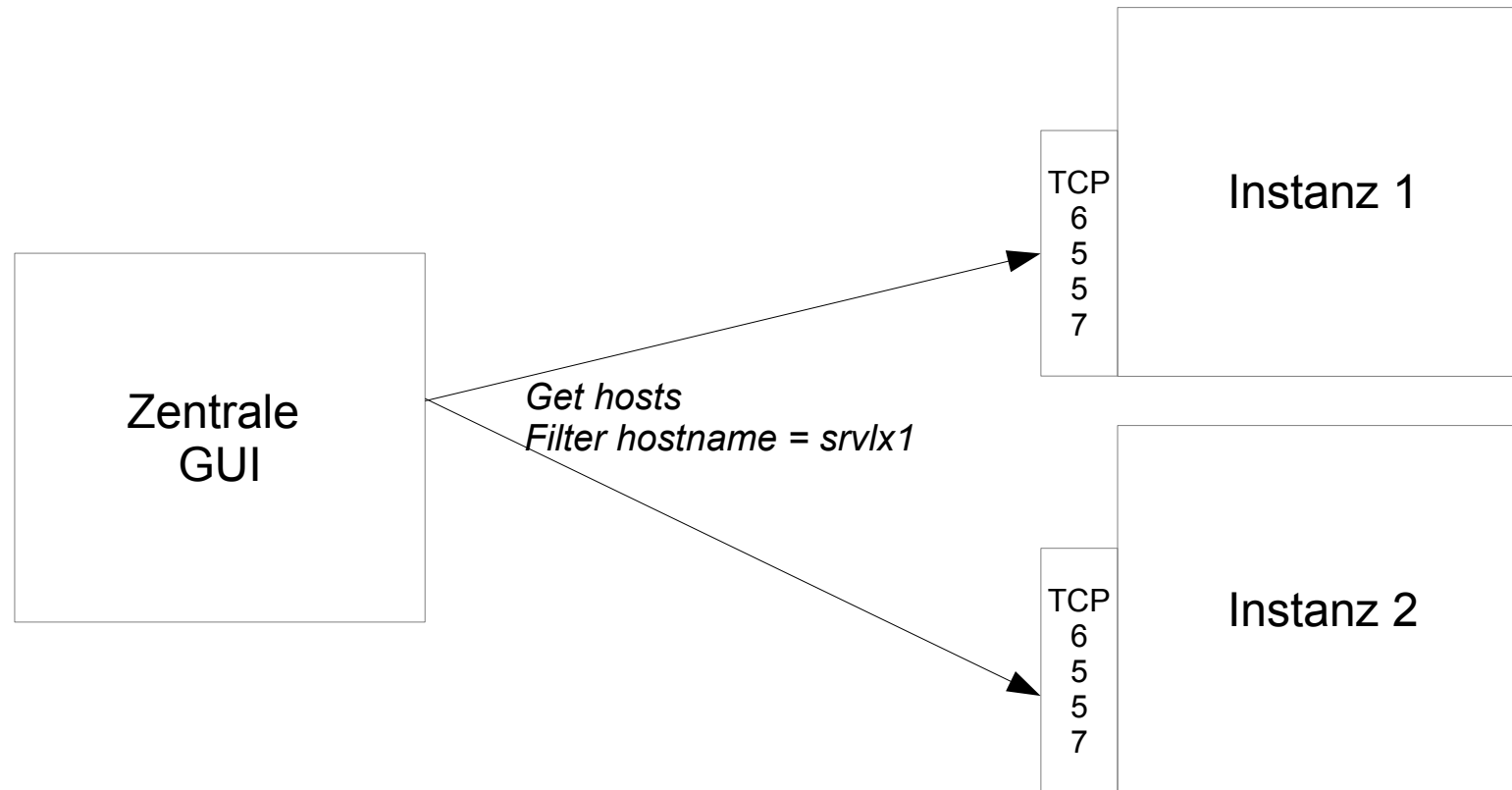


Multisite Livestatus

- Zentrale GUI
- evtl. zentrale Konfiguration
- Livestatus Socket per TCP als Backend

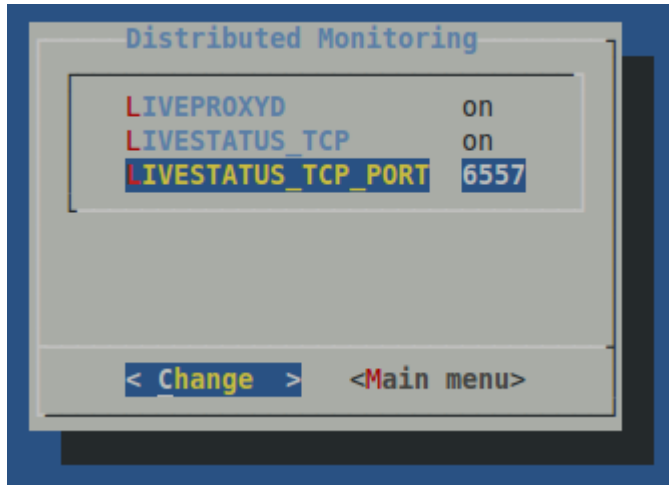


Multisite Livestatus



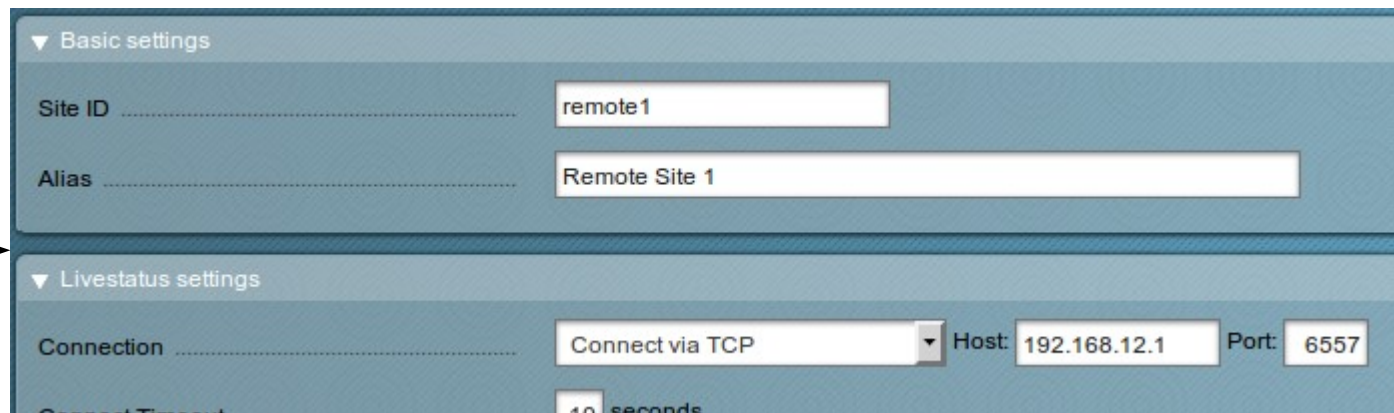


Einrichtung



← Aktivierung auf Instanz

Einrichtung auf Zentrale →





Vor- und Nachteile

Vorteile

- Sehr einfach zu konfigurieren
- Bei Verbindungsschwierigkeiten kein Datenverlust oder Verzögerung

Nachteile

- Skalierung nur bis etwa 50-60 Standorte
- Abfragegeschwindigkeit richtet sich nach langsamster Leitung



Kriterien

- Datenrichtung: von Master zur Slave Instanz
- Zuverlässigkeit: hoch
- Konf.aufwand: einmalig
- Skalierbarkeit: etwa 60 Instanzen
- Konsistenz: voll gegeben
- Funktioniert autonom



Kunden Screenshot

multisite
check_  1.1.12p1

Tactical Overview X

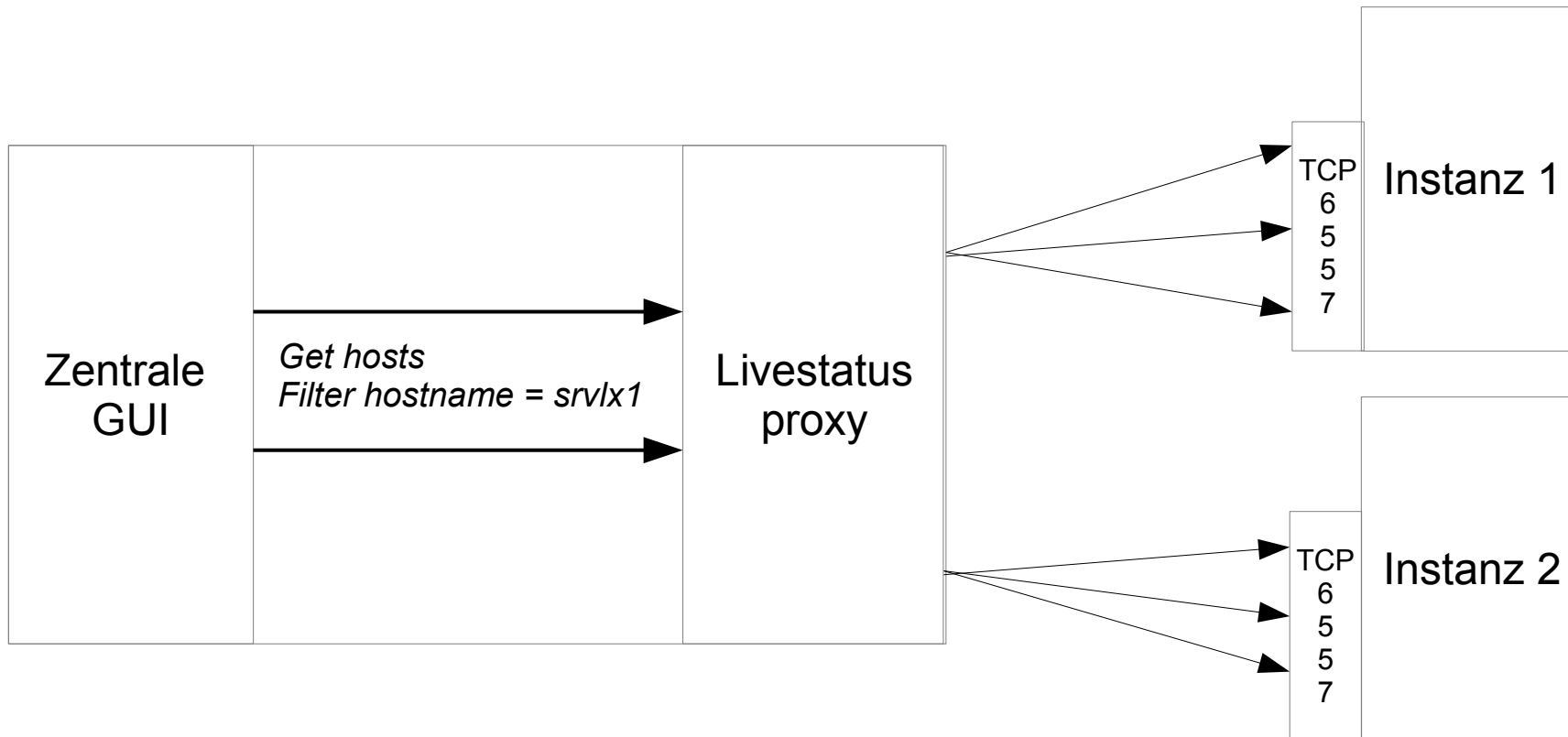
Hosts	Problems	Unhandled
32501	48	42
Services	Problems	Unhandled
1328682	947	529



- Sehr schnelles erkennen von schlechten Verbindungen
- Cachen von statischen Anfragen
- Aufrechterhalten von Verbindung -> viel kürzere Latenz!



Liveproxyd

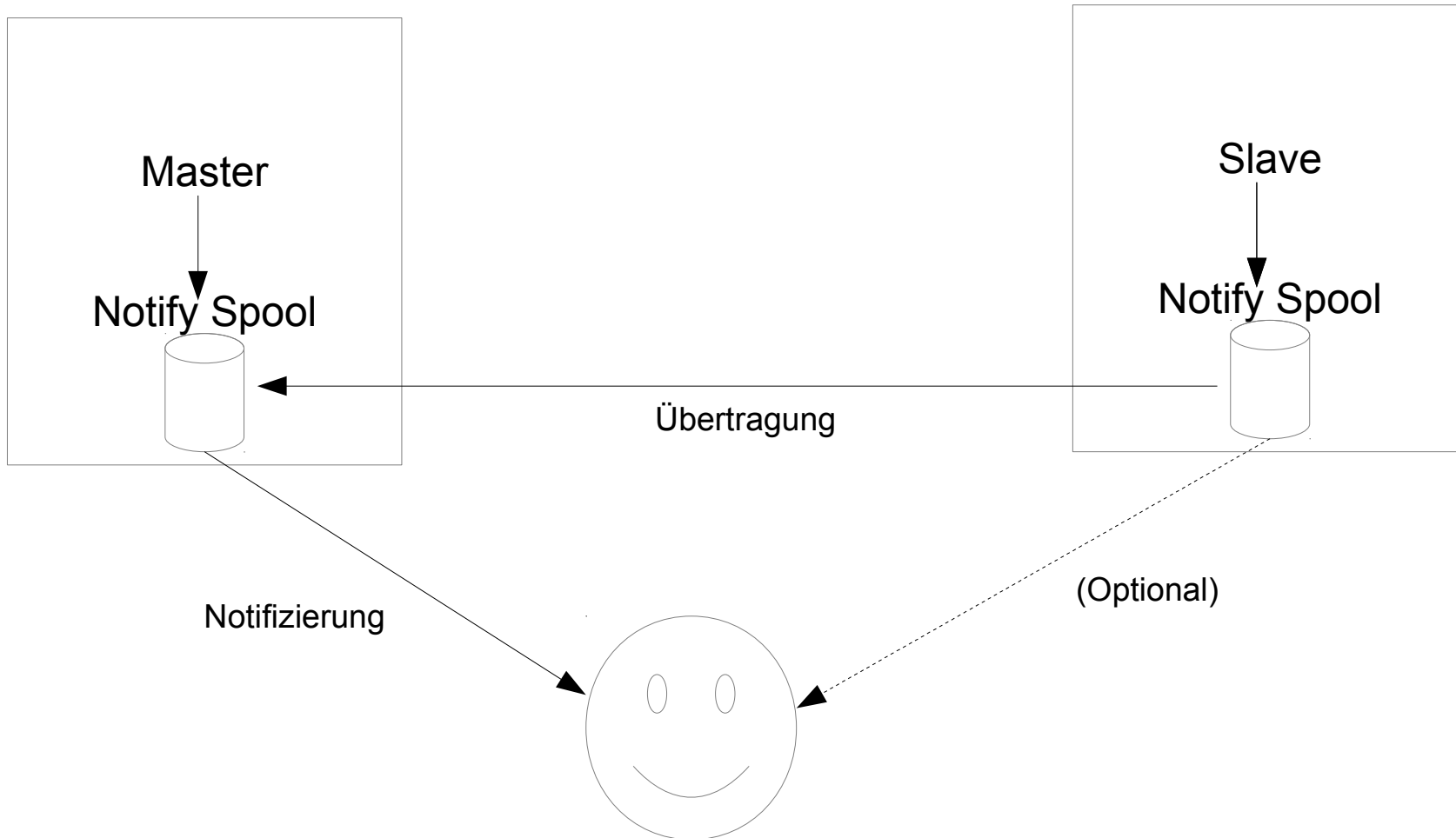




Notifikationen?



- Spoolt Benachrichtigungen zum Master
- Gleichzeitig Versand auch von Slave möglich

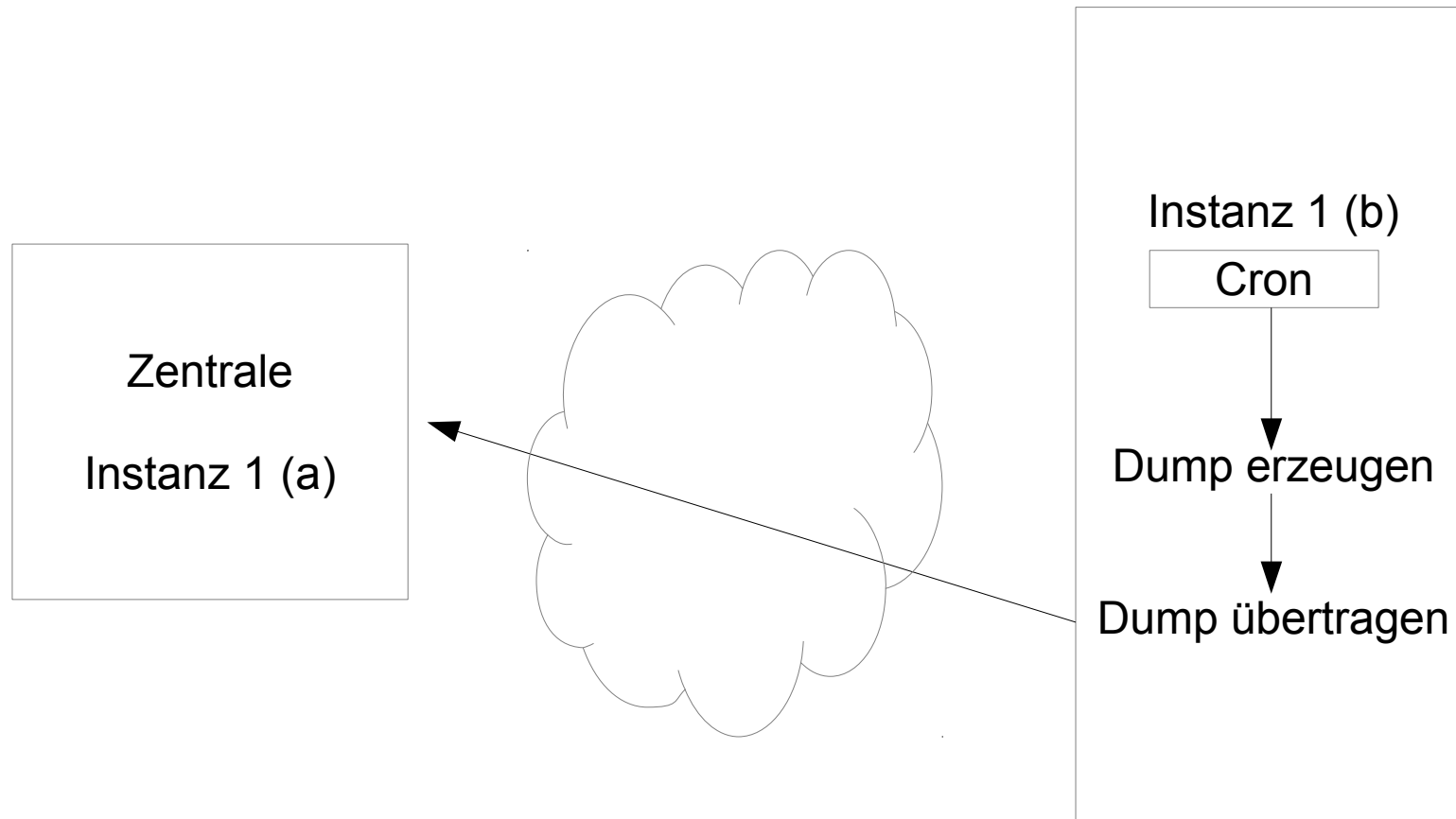




- Periodischer Abzug aller Monitoring Daten
- Übertragung der Daten auf beliebigen Weg zum Master
- Master Instanz wird mit Daten befüllt



Livedump





Vorteile

- Hohe Sicherheit: Zentrale braucht keinen Zugriff auf Remotesites

Nachteile

- Manuelle Einrichtung
- Acknowledgment in Zentrale auf Remotesite nicht sichtbar



Kriterien

- Datenrichtung: von Slave zur Master Instanz
- Zuverlässigkeit: muss überwacht werden
- Konf.aufwand: etwas erhöht
- Skalierbarkeit: abhängig von Servicezahl
- Konsistenz: nicht gegeben

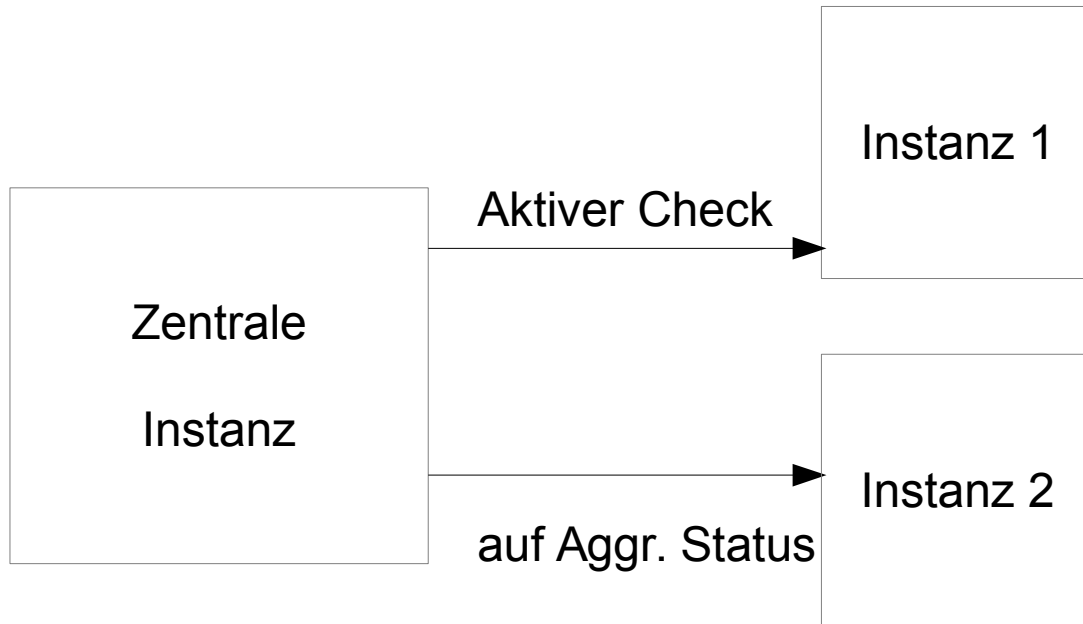


BI Aggregationen

- Aggregation frei definierbar
- Abfrage des Gesamtstatus aus Zentrale



Bi Aggregation



Links	State	Tree
	WA	▼ WebShop
	WARN	OK router ♦ Internet Connection ♦ OK - 127.17.44.1: rta 0.03
		OK ▶ Loadbalancers
		OK ▶ aggregating the single lines of WebShop
		WA mysqlserver ♦ MySQL Daemon Sessions ♦ Assumed to b



Vor- und Nachteile

Vorteile

- Skaliert bis zu vielen tausend Standorten
- Übersichtlich

Nachteile

- Ein aktiver Check pro Instanz
- Weniger Details direkt im Zentralsystem abrufbar



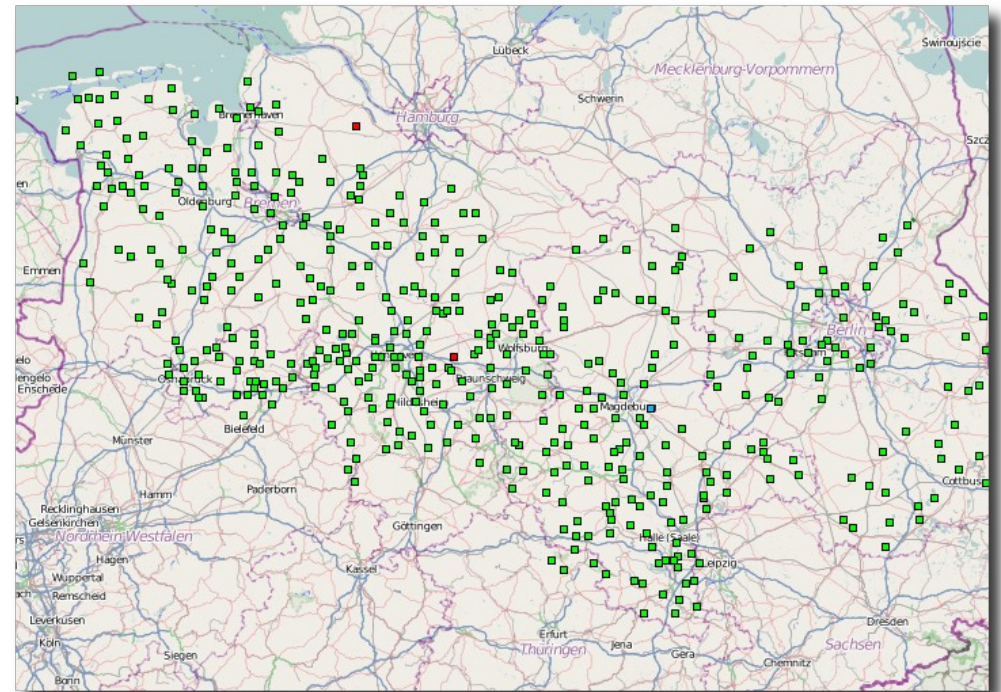
Kriterien

- Datenrichtung: von Master zur Slave Instanz
- Zuverlässigkeit: hoch
- Konf.aufwand: einmalig nötig
- Skalierbarkeit: mehrere tausend Instanzen
- Konsistenz: Systeme sind getrennt



Point of Sales Monitoring bei EDEKA

- Überwachung von 1.200 Filialen in Norddeutschland
- Autarkes Monitoring in den Filialen
- Zentrales Dashboard in Minden
- Rollout Februar 2012



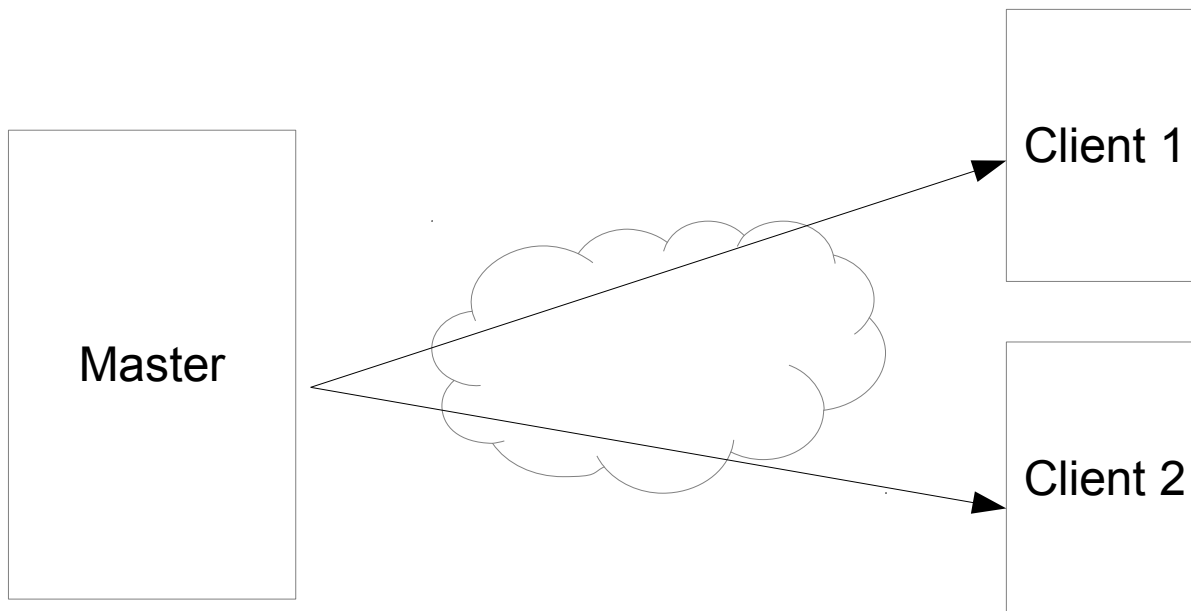


Ohne dezentralen Monitoringserver

- einfach die Checks übers Netzwerk laufen lassen
- Überwachen z.B. über SSH



Ohne dezentralen Monitoringserver





Vor- und Nachteile

Vorteile

- Einfach, da kein zusätzlicher Server nötig ist

Nachteile

- Höhere Laufzeiten
- ständiger Netzwerkverkehr durch Monitoring
- bei Leitungsausfall kein Monitoring mehr



Kriterien

- Datenrichtung: von Master zu Client
- Zuverlässigkeit: hoch
- Konf.aufwand: normal
- Skalierbarkeit: abhängig von Servicezahl
- Konsistenz: voll

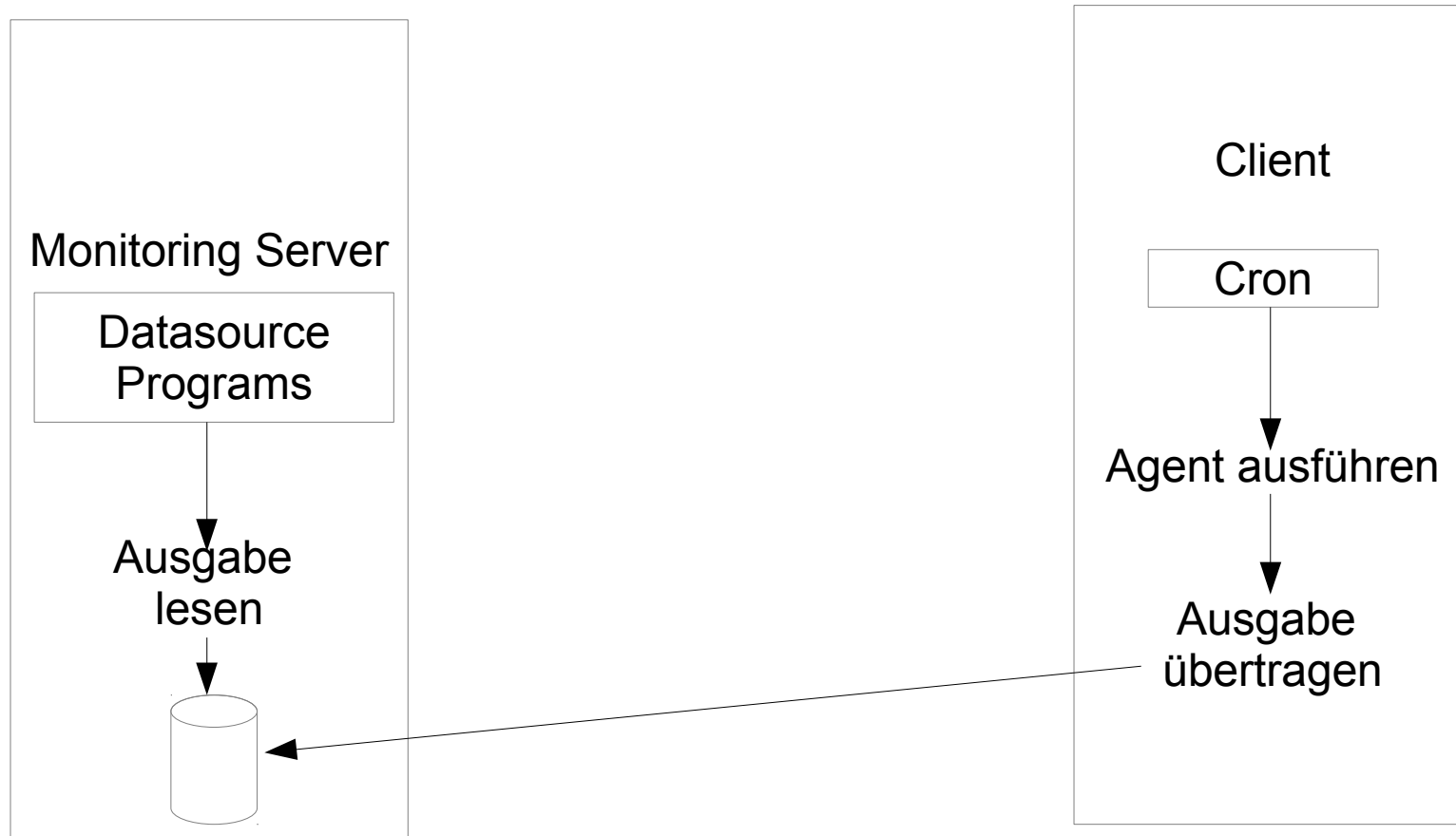


Passives übertragen

- Agenten Ausgabe auf Zentrale übertragen
- Push per SSH möglich



Passives übertragen





Vor- und Nachteile

Vorteile

- Sehr sicher, ähnlich wie Livedump
- keine Datenredundanz wie bei Livedump

Nachteile

- Konfigurationsaufwand
- Versenden der Daten muss selbst geskriptet werden
- geht nicht für SNMP



Kriterien

- Datenrichtung: von Client zu Master
- Zuverlässigkeit: verbindungsabhängig
- Konf.aufwand: einmalig
- Skalierbarkeit: abhängig von Servicezahl
- Konsistenz: voll



Was wir nicht verfolgen



- zentrale SQL-Datenbank
 - Sehr hoher Ressourcenverbrauch
 - Zusätzlicher Administrationsaufwand
 - skaliert schlecht
 - Evtl. Datenredundanz mit Remotesystemen
- NSCA
 - Hoher Konfigurationsaufwand



- mod_gearman
 - Wird von Check_MK nicht unterstützt
 - Gleiche Nachteile wie bei direktem Remote-Monitoring



Vielen Dank für Ihre Aufmerksamkeit