

Red Hat  
**Summit**

**Connect**

# Computer Vision per il controllo di qualità e la sostenibilità ambientale

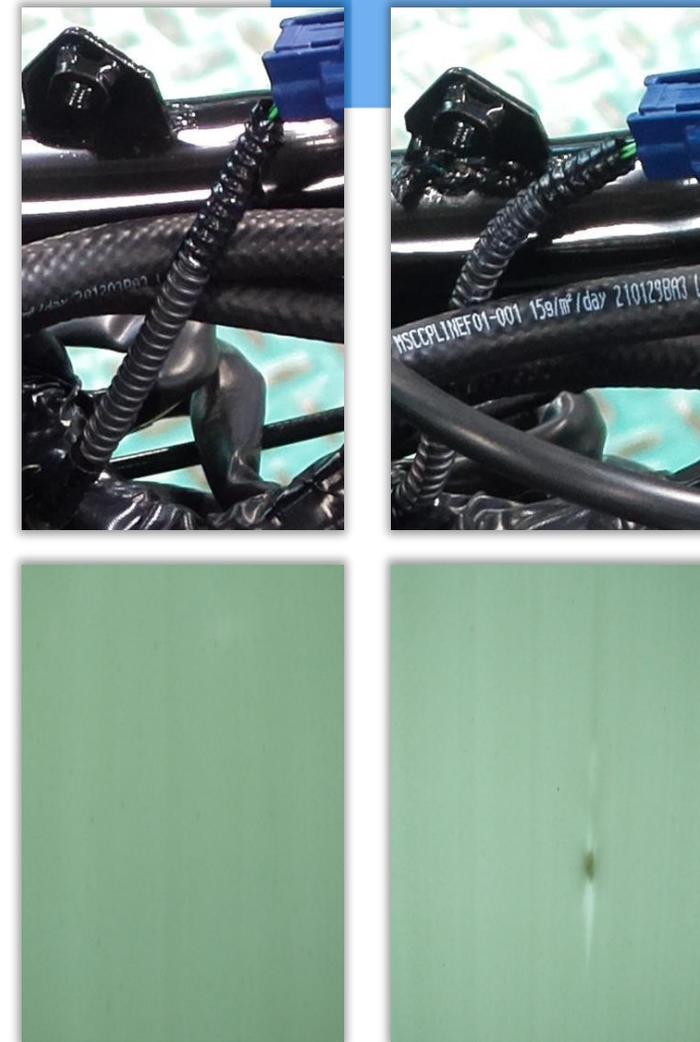
**Paolo Mascellani, CEO**  
eLabor

Milano, 19 novembre 2024



# Riconoscimento dei difetti

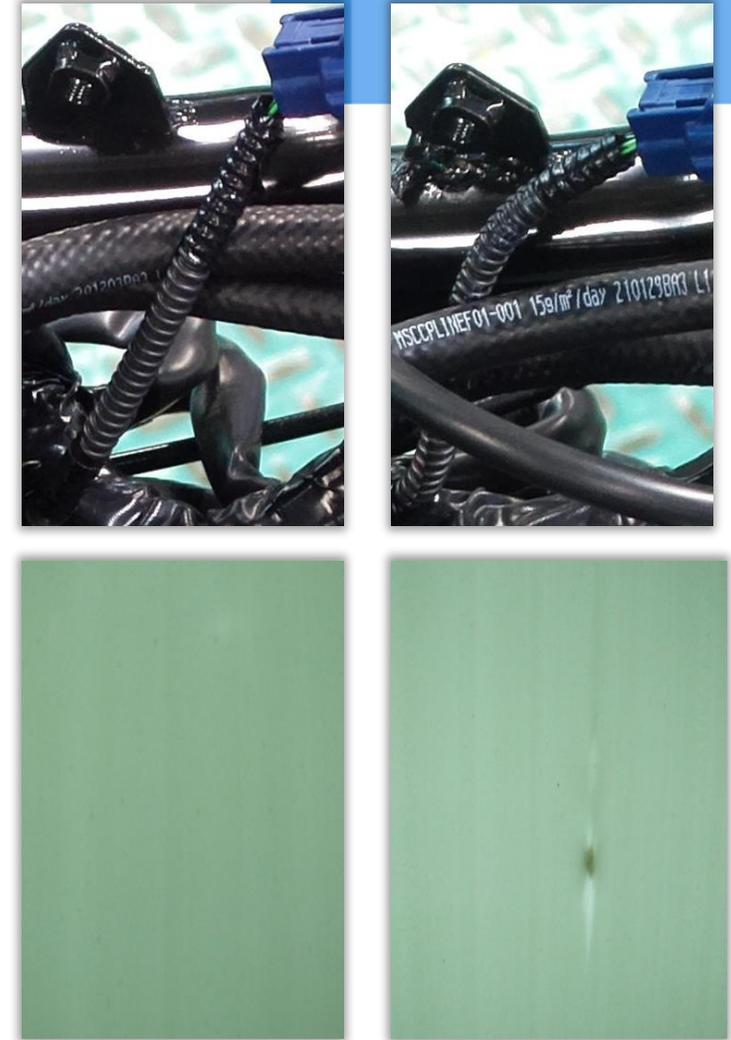
Molti difetti di produzione sono **riconoscibili a partire da immagini** (eventualmente anche radiografie, tomografie o altri tipi di immagini).



# Riconoscimento dei difetti

Molti difetti di produzione sono **riconoscibili a partire da immagini** (eventualmente anche radiografie, tomografie o altri tipi di immagini).

Questo è un tipico problema di **Computer Vision** per cui possono essere utilizzate tecniche di classificazione basate su processi di **Machine Learning** (usando ad esempio **Reti Neurali Profonde**).



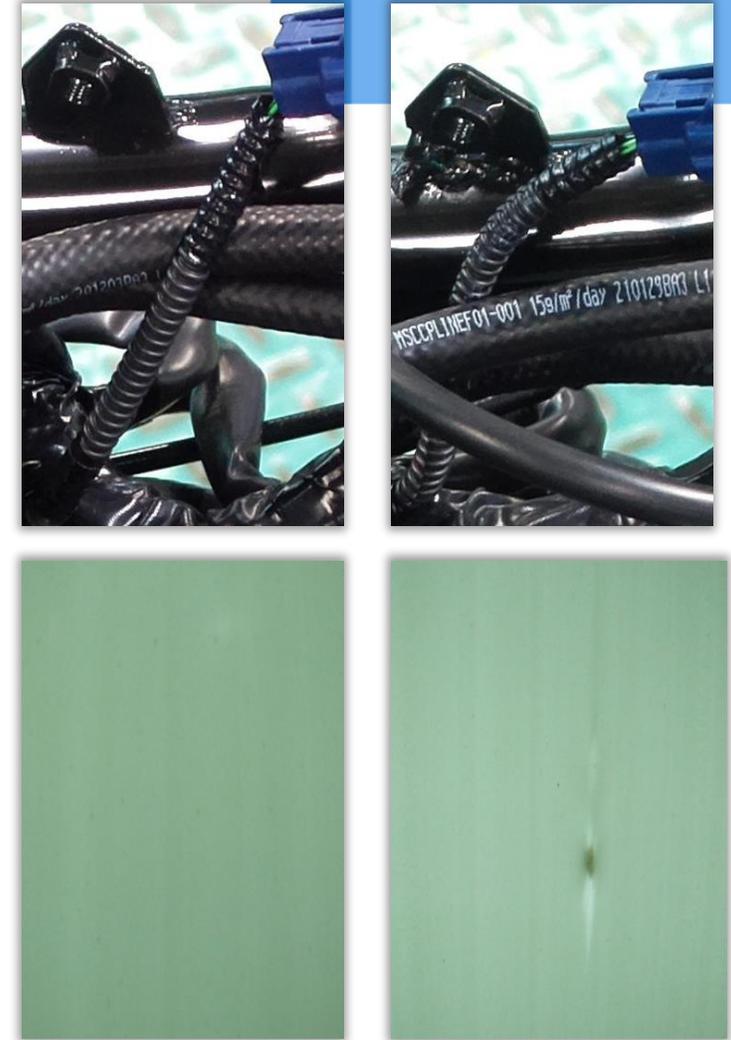
# Riconoscimento dei difetti

Molti difetti di produzione sono **riconoscibili a partire da immagini** (eventualmente anche radiografie, tomografie o altri tipi di immagini).

Questo è un tipico problema di **Computer Vision** per cui possono essere utilizzate tecniche di classificazione basate su processi di **Machine Learning** (usando ad esempio **Reti Neurali Profonde**).

Il flusso di lavoro tipico comprende:

- **acquisizione** immagini (anche più di una per campione) in **istanti ben precisi** (produzioni discrete) o a **intervalli regolari** (produzioni continue)
- **pre-elaborazione** immagini (resize, resample, rescale, crop, ... )
- **classificazione** delle immagini rispetto ai **diversi tipi di difetto** previsti
- **segnalazione dei difetti** rilevati (M2H / M2M)
- **memorizzazione** immagini e metadati



# Riconoscimento di oggetti

In molti casi, non si tratta di classificare l'immagine intera, ma di individuare se all'interno dell'immagine siano **presenti determinati tipi di oggetti** e in questo caso di **evidenziarli**.

Vogliamo ad esempio sapere se ci sono degli **intrusi** (sardine in una cassetta di alici?), oppure se è necessario **fare qualcosa di particolare** (un pedone sta attraversando la strada?)



# Riconoscimento di oggetti

In molti casi, non si tratta di classificare l'immagine intera, ma di individuare se all'interno dell'immagine siano **presenti determinati tipi di oggetti** e in questo caso di **evidenziarli**.

Vogliamo ad esempio sapere se ci sono degli **intrusi** (sardine in una cassetta di alici?), oppure se è necessario **fare qualcosa di particolare** (un pedone sta attraversando la strada?)

Anche questo è un tipico problema di Computer Vision per cui possono essere utilizzate **tecniche di segmentazione semantica**, anch'esse basate su processi di **Machine Learning** e **Reti Neurali Profonde**.

Il flusso di lavoro è molto simile al precedente (cambiano solo i task specifici): acquisizione, pre-elaborazione immagini, **identificazione delle istanze** dei vari tipi di oggetti previsti, **segnalazione delle istanze** rilevate, memorizzazione delle immagini e dei metadati.



# La soluzione

## Estrusore pilota

- Un **piccolo estrusore** con il quale vengono **testate** le partite di PE in ingresso
- Viene alimentato con un campione della partita e **valutata la qualità dell'estruso**
- Processo lungo ed oneroso (qualche ora)

## Potenziato da ADR-Flow

Il nostro sistema di Computer Vision per il **rilevamento automatico dei difetti** basato su Reti Neurali Profonde (DNN) e algoritmi di Machine Learning



# Il backstage

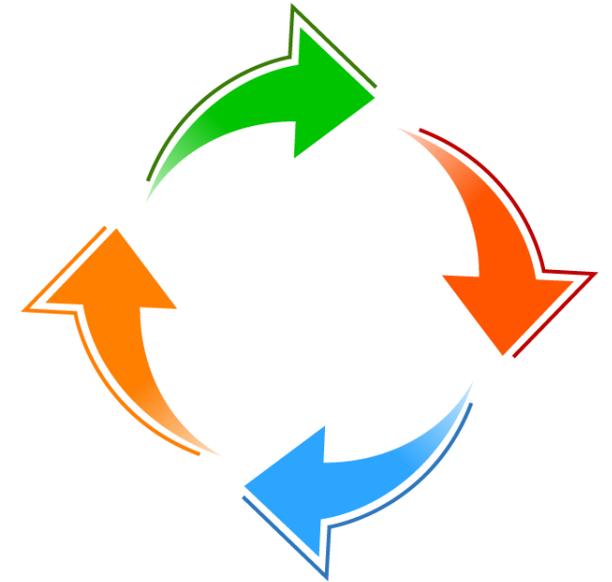
Per poter fare tutto questo bisogna avere **un'infrastruttura che supporti al meglio i task** che abbiamo visto e tutti quelli che servono di supporto:



# Il backstage

Per poter fare tutto questo bisogna avere **un'infrastruttura che supporti al meglio i task** che abbiamo visto e tutti quelli che servono di supporto:

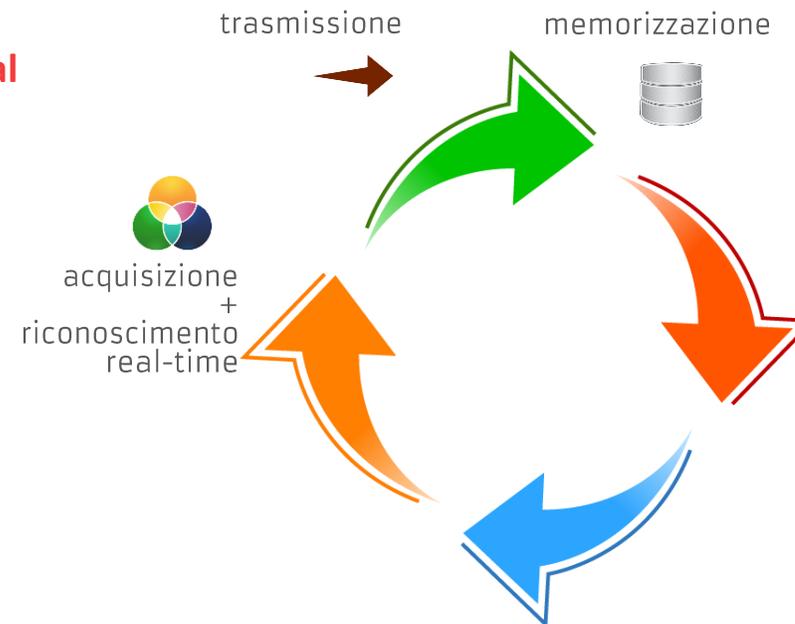
- infrastrutture di base: networking, storage (**immagini** e **metadati**), backup



# Il backstage

Per poter fare tutto questo bisogna avere **un'infrastruttura che supporti al meglio i task** che abbiamo visto e tutti quelli che servono di supporto:

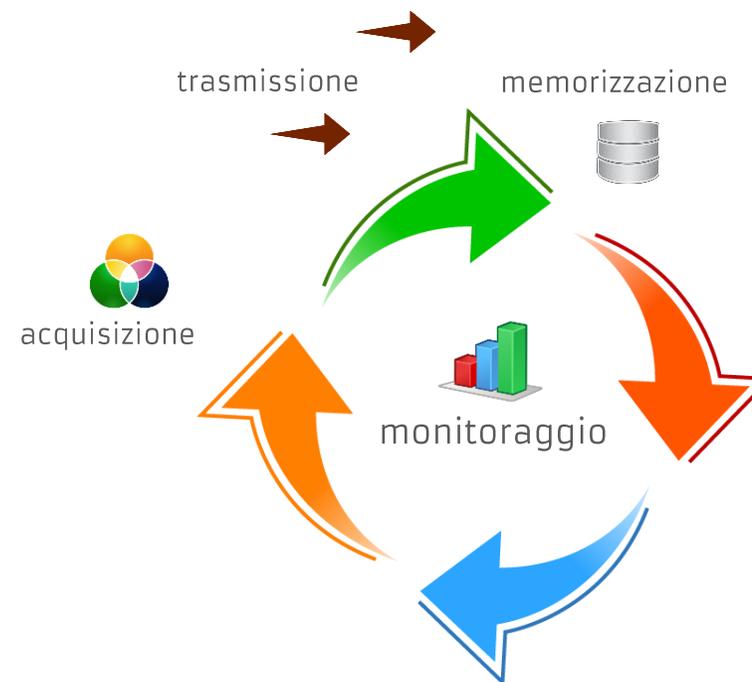
- infrastrutture di base: networking, storage (**immagini** e **metadati**), backup
- **acquisizione** e **streaming** dei dati



# Il backstage

Per poter fare tutto questo bisogna avere **un'infrastruttura che supporti al meglio i task** che abbiamo visto e tutti quelli che servono di supporto:

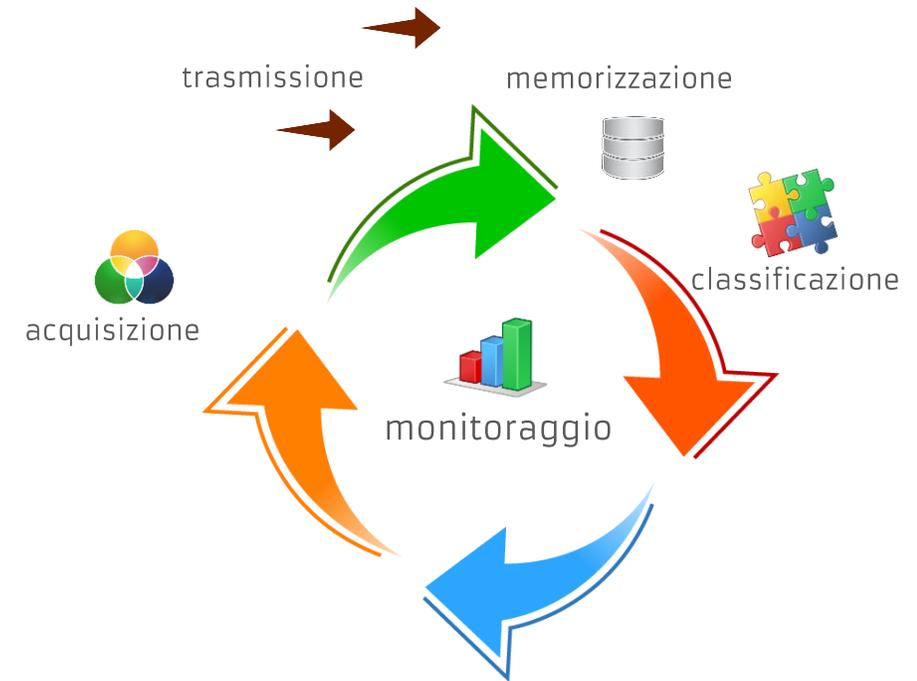
- infrastrutture di base: networking, storage (**immagini** e **metadati**), backup
- **acquisizione** e **streaming** dei dati
- **monitoraggio** e **dashboarding**



# Il backstage

Per poter fare tutto questo bisogna avere **un'infrastruttura che supporti al meglio i task** che abbiamo visto e tutti quelli che servono di supporto:

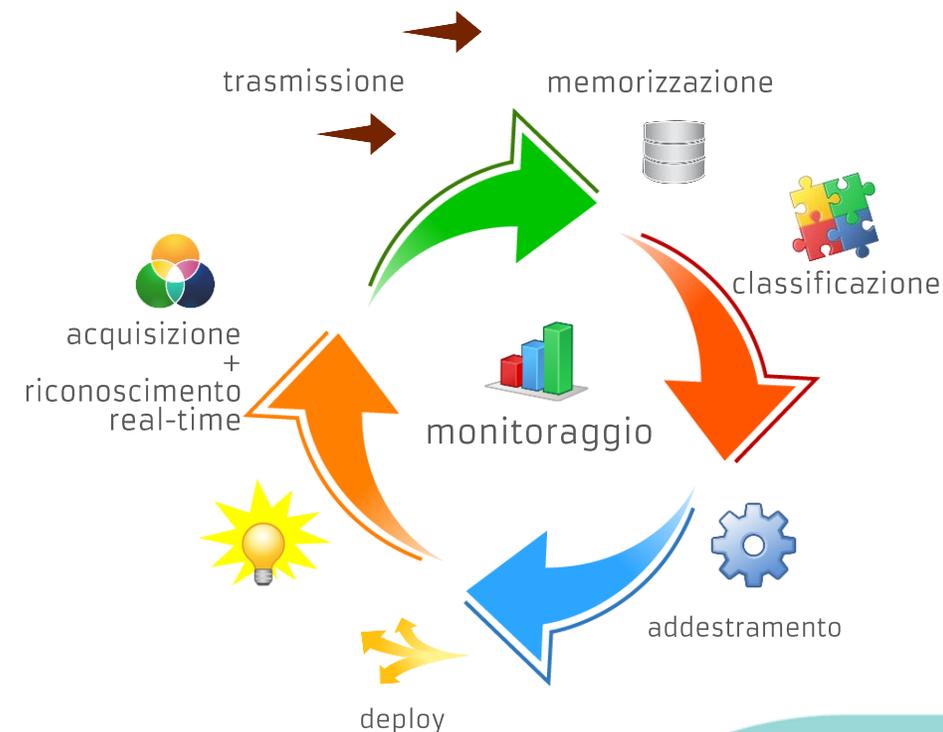
- infrastrutture di base: networking, storage (**immagini** e **metadati**), backup
- **acquisizione** e **streaming** dei dati
- **monitoraggio** e **dashboarding**
- interfacce utenti specifiche: **classificazione manuale** e **visual prompting**



# Il backstage

Per poter fare tutto questo bisogna avere **un'infrastruttura che supporti al meglio i task** che abbiamo visto e tutti quelli che servono di supporto:

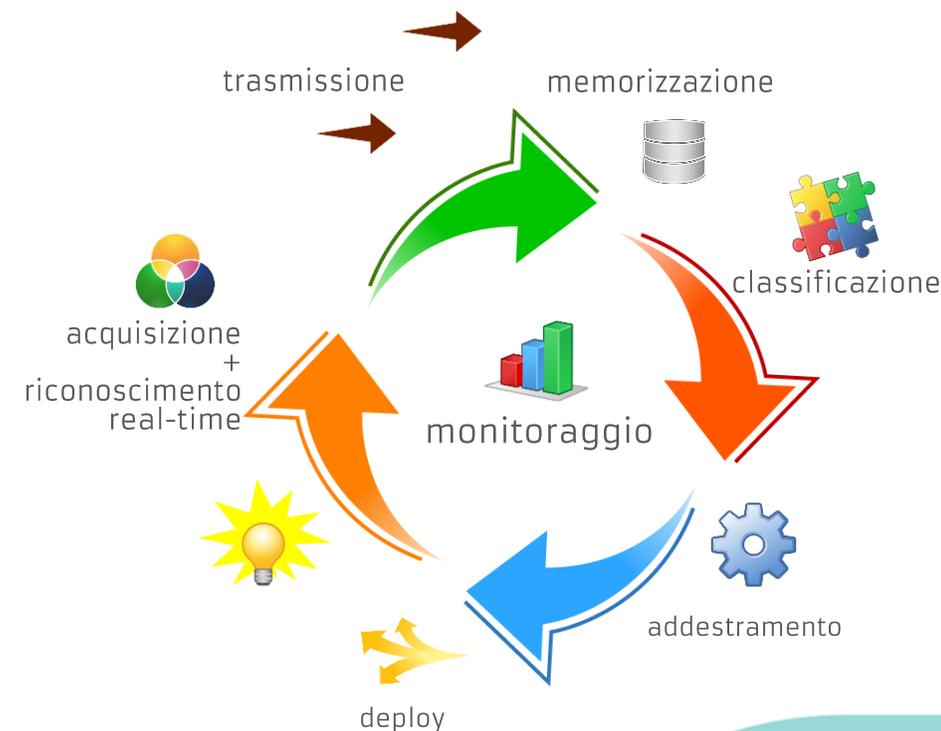
- infrastrutture di base: networking, storage (**immagini** e **metadati**), backup
- **acquisizione** e **streaming** dei dati
- **monitoraggio** e **dashboarding**
- interfacce utenti specifiche: **classificazione manuale** e **visual prompting**
- framework IA e MLOps: **addestramento**, **validazione**, **predizione**, modelli **pre-addestrati**, **gestione** e **deploy** dei modelli, **continuous testing**, **drift detection**, ...



# Il backstage

Per poter fare tutto questo bisogna avere **un'infrastruttura che supporti al meglio i task** che abbiamo visto e tutti quelli che servono di supporto:

- infrastrutture di base: networking, storage (**immagini** e **metadati**), backup
- **acquisizione** e **streaming** dei dati
- **monitoraggio** e **dashboarding**
- interfacce utenti specifiche: **classificazione manuale** e **visual prompting**
- framework IA e MLOps: **addestramento**, **validazione**, **predizione**, modelli **pre-addestrati**, **gestione** e **deploy** dei modelli, **continuous testing**, **drift detection**, ...
- Per tutte queste esigenze (e molte altre) esistono svariati strumenti **Open Source** che spesso possono essere considerati **standard di mercato** o comunque del tutto competitivi con i migliori sistemi proprietari.

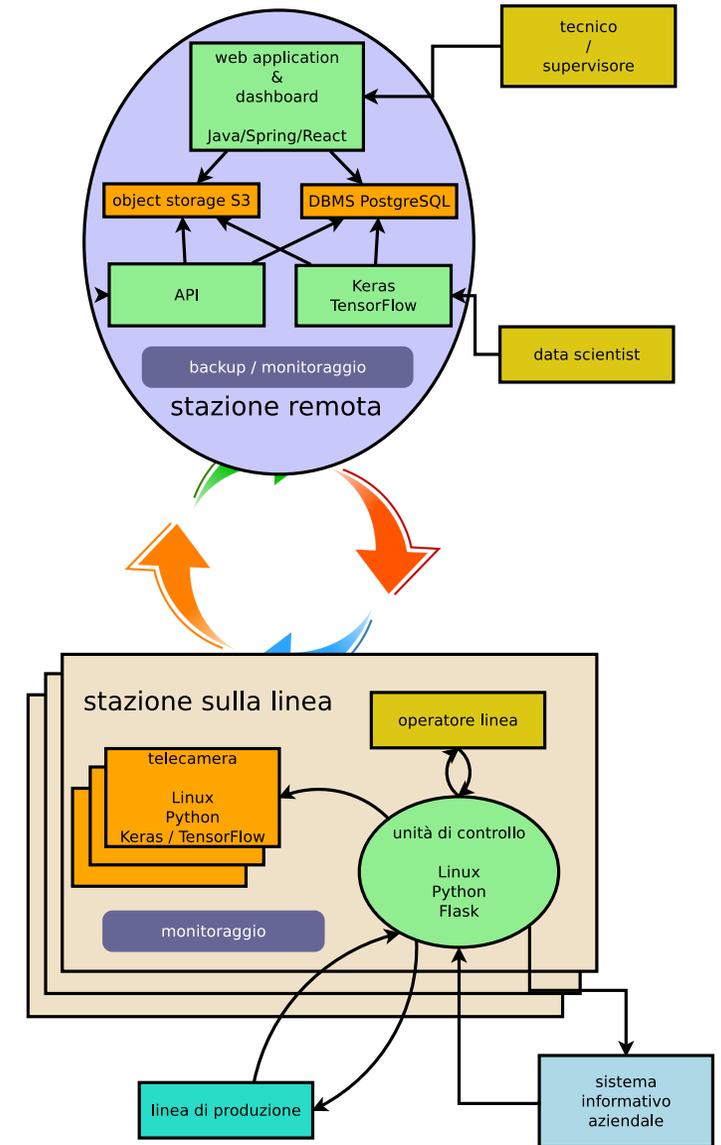


# Il backstage

Tutto questo è a disposizione di tutti coloro che vogliono **cimentarsi** nella costruzione di un sistema di Computer Vision, così come corsi, tutorial, esempi, e tutto quello che serve, ma ... **non tutti sono specialisti di IA** e magari non intendono neppure diventarlo (anche se, almeno un po', sarà probabilmente necessario).

Per questo abbiamo costruito **ADR-Flow**: un sistema software Open Source ed un'infrastruttura di riferimento per sistemi di Computer Vision

- dedicata soprattutto ai **sistemi industriali** o comunque **produttivi**, ma **utilizzabile in molti altri contesti**
- che **integra** (solo) strumenti Open Source
- in modo **adattabile** e **flessibile**
- così da permettere a chiunque di **concentrarsi sulla sua specifica esigenza**, evitando di affrontare i problemi di integrazione ed infrastrutturali



Red Hat  
**Summit**

**Connect**

Q&A



Red Hat  
**Summit**

**Connect**

**Grazie!**



[paolo@elabor.biz](mailto:paolo@elabor.biz)  
<https://elabor.biz>  
<https://youtube.elabor.biz>  
<https://linkedin.elabor.biz>

